

Code Inspection Instructional Validation

**Kay Orr
Shirley Stancil**

Southwest Research Institute

July 30, 1992

N92-33411

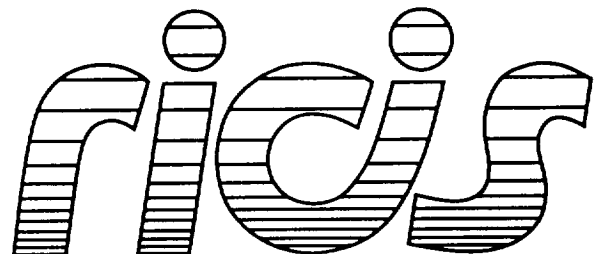
Unclas

G3/61 0116928

**Cooperative Agreement NCC 9-16
Research Activity No. ET.31**

**NASA Johnson Space Center
Engineering Directorate
Flight Data Systems Division**

**(NASA-CR-190718) CODE INSPECTION
INSTRUCTIONAL VALIDATION (Research
Inst. for Computing and Information
Systems) 152 p**



*Research Institute for Computing and Information Systems
University of Houston-Clear Lake*

TECHNICAL REPORT

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

Code Inspection ***Instructional Validation***

RICIS Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Dr. Kay Orr and Shirley Stancil of the Southwest Research Institute. Dr. Glenn Freedman served as the RICIS research coordinator.

Funding was provided by the Engineering Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA research coordinator for this activity was William R. Pruett of the Software Integration and Maintenance Section of the Flight Data Systems Division, Engineering Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the authors and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, NASA or the United States Government.

EXECUTIVE SUMMARY

Objectives of the Study

The Shuttle Data Systems Branch (SDSB) of the Flight Data Systems Division (FDSD) at Johnson Space Center contracted with Southwest Research Institute (SwRI) to validate the effectiveness of an interactive video course on the code inspection process. The purpose of this project was to determine if this course could be effective for teaching NASA analysts the process of code inspection. In addition, NASA was interested in the effectiveness of this unique type of instruction (Digital Video Interactive®), for providing training on software processes.

Conclusions

This study found the Carnegie Mellon course, "A Cure for the Common Code", effective for teaching the process of code inspection. In addition, analysts prefer learning with this method of instruction, or this method in combination with other methods. As is, the course is definitely better than no course at all; however, findings indicate changes are needed. Following are conclusions of this study:

- The course is instructionally effective.
- The simulation has a positive effect on student's confidence in their ability to apply new knowledge.
- Analysts like the course and prefer this method of training, or this method in combination with current methods of training in code inspection, over the way training is currently being conducted.
- Analysts responded favorably to information presented through scenarios incorporating full motion video.
- Some course content needs to be changed.
- Some content needs to be added to the course.

Recommendations

SwRI believes this study indicates interactive video instruction combined with simulation is effective for teaching software processes. Based on the conclusions of this study, SwRI has outlined seven options for NASA to consider. SwRI recommends the option which involves creation of new source code and data files, but uses much of the existing content and design from the current course. Although this option involves a significant software development effort, SwRI believes this option will produce the most effective results.

TABLE OF CONTENTS

	Page
1.0 INTRODUCTION	1
1.1 Purpose	1
1.2 Overview of the Study	1
1.2.1 Methodology	1
1.2.2 Findings	1
1.2.3 Conclusions	2
1.2.4 Limitations	2
1.2.5 Recommendations	2
2.0 METHODOLOGY	3
2.1 Approach of the Study	3
2.1.1 Validate Content of the Course	3
2.1.1.1 Compare Code Inspection Objectives	3
2.1.1.2 Compare Models for Code Inspection	3
2.1.2 Validate Effectiveness of the Course	3
2.1.2.1 Knowledge of Information	4
2.1.2.2 Application of Information	4
2.1.2.3 Analyst Opinions	4
2.2 Subjects	4
2.3 Materials	5
2.3.1 Carnegie Mellon Course	5
2.3.2 Instruments	5
2.3.2.1 Pretest/Posttest	5
2.3.2.2 Analyst Questionnaire	6
2.3.2.3 Interview Questions	6
2.3.2.4 Observation Form	6
2.3.2.5 Demographic Data Sheet	7
2.4 Procedures	7
2.4.1 Study Procedures	7
2.4.2 Data Collection Procedures	7
3.0 FINDINGS	8
3.1 Results of Data Collection	8
3.1.1 Content Review Findings	8
3.1.1.1 Content Review by SwRI	8
3.1.1.2 Content Review by IBM	8
3.1.1.3 Code Inspection Model Review by SwRI	9
3.1.1.4 Code Inspection Model Review by IBM	9
3.1.2 Instructional Effectiveness Review Findings	10
3.1.2.1 Pretest/Posttest 1 (Knowledge of Information)	10
3.1.2.2 Posttest 1/Posttest 2 (Application of Information)	10

3.1.2.3	Program Feedback	11
3.1.2.4	Analyst Self Report	11
3.1.2.4.1	Analyst Questionnaire	11
3.1.2.4.2	Interview	12
3.1.2.5	Evaluator Observations	13
3.2	Summary of Findings	14
3.2.1	Strengths of the Course	14
3.2.1.1	Instructional Issues	14
3.2.1.2	Aesthetic Issues	14
3.2.1.3	Technical Issues	14
3.2.2	Weaknesses of the Course	14
3.2.2.1	Instructional Issues	14
3.2.2.2	Aesthetic Issues	15
3.2.2.3	Technical Issues	15
4.0	CONCLUSIONS	16
4.1	Conclusions of the Study	16
4.1.1	Content	16
4.1.2	Instructional Effectiveness	16
4.1.3	Analyst Opinions	17
5.0	LIMITATIONS	18
5.1	Limitations of the Study	18
5.2	Limitations of the Course	18
5.3	Limitations of the Course as cited by Carnegie Mellon	18
5.3.1	General	18
5.3.2	Training Room	19
5.3.3	Library	19
5.3.4	Conference Room	19
6.0	RECOMMENDATIONS	20
6.1	Conclusion Summary	20
6.2	NASA Options	20
6.3	SwRI Recommendation	23
6.3.1	Modification Suggestions	24
6.3.1.1	Content	24
6.3.1.2	Presentation	24
6.3.1.3	User Interface Features	24
7.0	REFERENCES	26

LIST OF APPENDICES

Appendix A Instruments

- Pretest/Posttest**
- Analyst Questionnaire**
- Interview Questions**
- Observation Form**
- Demographic Data Sheet**
- Instructional Path**

Appendix B Raw Data

- Program Feedback**
- Analyst Questionnaire**
- Interview Questions**
- Observations**
- Demographic Information**

Appendix C Annotated Course Objectives

Appendix D Carnegie Mellon Course Objectives

Appendix E Carnegie Mellon Code Inspection Model

Appendix F IBM Comparison of Objectives and Code Inspection Models

Appendix G Validation Plan

Appendix H Outline of "A Cure for the Common Code"

1.0 INTRODUCTION

1.1 Purpose

Analysts in the NASA Flight Data Systems Division (FDSD) at Johnson Space Center manage the software configuration for the maintenance of shuttle software. NASA management recognizes the need for effective, efficient training to provide these analysts with the knowledge and skills necessary to perform their jobs.

The purpose of this study was to determine the effectiveness of an interactive video course on the code inspection process. "A Cure for the Common Code", developed by the Software Engineering Institute at Carnegie Mellon University, consists of training modules, a reference library and simulated code inspections. In addition to determining the effectiveness of this particular piece of courseware, management is also interested in the effectiveness of this unique type of instruction (DVI®) for teaching content relevant to NASA needs. To investigate these issues, the NASA Shuttle Data Systems Branch (SDSB) contracted with the Training Systems and Simulators Department at Southwest Research Institute (SwRI) to conduct this study.

1.2 Overview of the Study

1.2.1 Methodology

SwRI created a plan to validate the effectiveness of the Carnegie Mellon code inspection course. The plan consisted of two parts: validation of the course content and validation of the instructional effectiveness of the course. Validation of the content was achieved by comparing code inspection objectives and comparing code inspection models. Validation of the effectiveness of the course was measured by testing knowledge of information, application of information, and by gathering analysts' reactions to the course.

Three analysts participated in the study, one from NASA and two from IBM. The analysts' backgrounds were varied in terms of computer experience, computer use, programming experience, experience with code inspection, and job responsibilities. Analysts with diverse backgrounds were sought in order to gather different perspectives on the code inspection course.

The materials used for this validation study included the inspection course, "A Cure for the Common Code". To measure the effectiveness of the course, SwRI developed instruments including: pretest/posttest, analyst questionnaire, interview questions, observation form, and demographic data sheet.

1.2.2 Findings

Findings for the study are presented as content review findings and instructional effectiveness review findings. Content review findings state results from the review of course objectives, course content and underlying process models for code inspection. Instructional effectiveness review findings report results in terms of gain scores between the pretest and Posttest 1 (knowledge of information), gain scores between Posttest 1 and Posttest 2 (application of information), program feedback, analyst responses to the questionnaire and interview, and evaluator observations. Finally, a summary of findings is presented in terms of strengths and weaknesses of the course.

1.2.3 Conclusions

Conclusions based on the findings are presented. Conclusions are presented for course content, instructional effectiveness of the course, and analyst opinions about the course.

1.2.4 Limitations

There were known limitations for this study which may have affected results. Limitations of the study are presented, as well as limitations for the course ("A Cure for the Common Code"). Limitations of the course as cited in documentation by Carnegie Mellon are also included.

1.2.5 Recommendations

Based on the conclusions of this study, SwRI has outlined seven options for NASA to consider. These options are presented along with pros and cons for each. SwRI recommends one option believed to be the most instructionally effective and most cost effective method for incorporating process simulation training into current training efforts.

2.0 METHODOLOGY

2.1 Approach of the Study

The plan created by SwRI to validate the effectiveness of Carnegie Mellon's code inspection course consisted of two parts. The objective of the first part was to validate the content of the course. The objective of the second part was to validate the instructional effectiveness of the course (including presentation strategy). Below is a summary of the validation plan. Appendix G contains a copy of the plan.

2.1.1 Validate Content of the Course

Content validity of the code inspection course was measured in two ways. First, instructional objectives were compared and second, models for code inspection were compared.

2.1.1.1 Compare Code Inspection Objectives

NASA objectives for the code inspection process, as indicated by descriptions in the Software Formal Inspections Guidebook (August, 1991) and the NASA Software Inspection Process Standard (December 9, 1991), were compared with Carnegie Mellon course objectives.

SwRI analysts took high-level objectives provided by Carnegie Mellon (see Appendix D) and added a fine level of detail from the course. SwRI took these annotated course objectives (see Appendix C) and compared them with the description of code inspection provided in the NASA documents. These same annotated objectives were given to IBM who, based on their experience, also compared them with the NASA code inspection process.

2.1.1.2 Compare Models for Code Inspection

Second, models for code inspection were compared. The model for code inspection used by NASA (Software Formal Inspections Guidebook and NASA Software Inspection Process Standard) was compared with the model used by Carnegie Mellon in the code inspection course (see Appendix E). Again, this comparison was done by both SwRI and verified by IBM.

2.1.2 Validate Effectiveness of the Course

Validation of instructional effectiveness involved three analysts using the code inspection course. Each analyst spent approximately six hours during one day working through the course (including tests, questionnaire and interview). SwRI chose three indicators of effectiveness for the code inspection course:

- knowledge of information
- application of information
- reactions to the course

2.1.2.1 Knowledge of Information

Analysts were given a pretest prior to receiving instruction. Upon completion of the course instructional modules, analysts were given a posttest. The purpose of this posttest was to measure knowledge of information as a result of having completed the instructional modules.

2.1.2.2 Application of Information

The course also simulates a code inspection. Analysts were given the opportunity to assume a role (recorder) and participate in a simulated code inspection. They were called upon to apply the knowledge and skills learned from the instructional modules in the training room. Course feedback was given to indicate analysts' ability to apply the information learned. In addition, a second posttest, given after completion of the simulation, measured any net gain or loss of information that may have resulted from the simulated experience.

2.1.2.3 Analyst Opinions

After completion of the entire code inspection course, analysts were administered a questionnaire and were then interviewed. The purpose of these two activities was to determine the analysts' reactions to the course, including their likes, dislikes, and suggestions for improvement and use.

2.2 Subjects

Three analysts participated in the study, one from NASA (FDSD) and two from IBM. Following is a summary of the analysts' backgrounds. A complete description of demographic information on the three analysts is provided in Appendix B.

Each analyst had between five and fourteen years of computer experience with between one and six hours of use per day. Computer usage included word processing, programming, and other applications. Each analyst had taken from one to seven college level software courses. None of the analysts had received any formal college instruction in the Ada programming language; however, other languages included Pascal, C, Cobol, LSP, Basic, Fortran, and informal instruction in Ada. Only one analyst had participated in a code inspection before, but two analysts indicated their job may require it in the future. None of the three analysts had ever received formal training in the code inspection process.

All three analysts had some experience with training via computer in the past, ranging from one to eight courses. They all felt it was an effective method for learning and had a positive attitude toward it. Interaction was cited as a major advantage of computer based training, as well as the feeling that it was much more interesting than reading from a manual.

2.3 Materials

2.3.1 Carnegie Mellon Course

The Carnegie Mellon course, "A Cure for the Common Code", is based on a fictitious company named "Ultimex". The course teaches the process of code inspection as if the student is a new employee. The course uses full motion video, still video, audio, and simulation.

Various rooms in the company are available to the user. The training room consists of five instructional modules:

- Formal Inspections: Purpose and Process
- Inspection Types and Differences
- Inspection Roles and Pitfalls
- Inspection Tools and Forms
- Inspection Communications

The conference room is where simulated code inspections occur. The code inspection simulation allows the student to apply what he/she learned from the instructional modules. The simulation uses a rule-based expert system of approximately one hundred rules. The expert system determines the responses of the simulated personalities, controls dialogue, interprets user responses, and controls the visual presentation.

Other rooms in the company are also available to the user. An overview of the course is given in the auditorium. A library is available for reference, which includes articles and manuals, as well as videotapes. The user has an office with tools available for reviewing code in preparation for a code inspection. Finally, to make the environment more realistic, the secretary's office and coffee room are also included. An outline of the course can be found in Appendix H.

2.3.2 Instruments

In addition to the course, a number of other materials were used in this study. Five instruments were created by SwRI to measure the effectiveness of the course, as well as reactions of the analysts participating in the study. The following instruments were created: pretest/posttest, analyst questionnaire, interview questions, an observation form, and a demographic data sheet. Brief descriptions of these instruments follow. Samples of these instruments appear in Appendix A.

2.3.2.1 Pretest/Posttest

The pretest and posttest (Posttest 1 and Posttest 2) were the same instrument. They were created by an instructional designer with pretest trials using three subjects that represented the target audience (software engineer, mathematician, engineer). The pretest trial results were reviewed by a subject matter expert (software engineer).

The purpose of the pretest/posttest was determined by when it was administered in the study. The same instrument was used three times, since time and resource constraints did not permit creation of different versions of the test.

When administered prior to the study (pretest), the instrument served to give a baseline measure of knowledge of information, so any gain in score after instruction could be measured. When used as a posttest for the first time (Posttest 1, after the instructional modules), the instrument served as a measure of knowledge of information. When used as a posttest for the second time (Posttest 2, after the simulated code inspection), the instrument measured gain in knowledge as a result of application. The purpose was to determine if analysts learned any more from the simulation.

The pretest/posttest instrument had three parts including definitions, fill in the blank, and multiple choice. Part 1, consisting of five questions, required analysts to define terms related to the code inspection process. Part 2, consisting of twenty fill in the blank questions, required analysts to recall information presented in the course. Part 3, consisting of forty multiple choice questions, required analysts to choose the correct answer(s) from the choices provided.

2.3.2.2 Analyst Questionnaire

The analyst questionnaire was administered after completion of the course. The purpose of the questionnaire was to measure analysts' reactions to the course and recommendations for change and use.

The analyst questionnaire consisted of a total of thirty-eight statements. These statements were grouped into five categories including overall evaluation of the course (nine statements), course content (five statements), learning effectiveness (two statements), instructional presentation (twelve statements), and system capabilities (ten statements). Analysts were asked to rate each statement on a scale of one to five (one indicating strong disagreement and five indicating strong agreement with the statement).

2.3.2.3 Interview Questions

The interview was conducted after completion of the course and immediately after the analyst questionnaire. The purpose of the interview was to further expand on information collected by the questionnaire regarding analysts' reactions to the course and recommendations for change and use.

Interview questions were grouped using the same five categories as the analyst questionnaire: overall evaluation of the course (ten questions), course content (seven questions), learning effectiveness (three questions), instructional presentation (four questions), and system capabilities (two questions). A total of twenty-six questions were included in the interview.

2.3.2.4 Observation Form

The observation form was used during data collection when analysts were actually using the course. The purpose of the observation form was to note any comments or actions during the use of the course which might be used to explain findings. During the entire use of the course, SwRI observers noted observations in four areas: problems or difficulties experienced, analyst comments, observer comments, time spent working on each task (modules, tests, simulation, etc.).

2.3.2.5 Demographic Data Sheet

The demographic data sheet was administered by NASA prior to data collection. The purpose of the demographic data sheet was to aid in choosing analysts to participate in the study, as well as helping explain findings.

The data sheet consisted of a total of sixteen questions grouped into four categories. The four categories of questions were computer experience (three questions), programming experience (three questions), code inspection process (seven questions), and general information (three questions).

2.4 Procedures

2.4.1 Study Procedures

At the beginning of this study, the code inspection course was obtained from Carnegie Mellon and loaded on an SwRI computer system for evaluation. Carnegie Mellon provided high-level instructional objectives and extensive detail was added by SwRI to create a set of annotated objectives.

The validation plan (see Appendix G) was written and dictated how the study would proceed. The plan consisted of two phases. Phase one called for assessment of the content validity of the course. Phase two called for three analysts to use the course, so data could be collected on effectiveness of the course, as well as analyst reactions to the course. The steps used in data collection are outlined below.

2.4.2 Data Collection Procedures

Prior to conducting trial runs of the course, NASA administered the demographic data sheet and the pretest to seven analysts (3 NASA analysts, 4 IBM analysts). Based on the information collected, SwRI chose three analysts (1 NASA analyst, 2 IBM analysts) to participate in the study. The proposal dictated the number of individuals that would participate in the study (three).

During the study, analysts were instructed how to proceed through the course. First, they were introduced to the course in the auditorium. Next, they participated in the five instructional modules in the training room. Upon completion of the modules, they were administered Posttest 1 to measure their knowledge of the information presented. After the posttest, analysts were allowed to explore the library, practice with the Ada code in their office, and finally participate in a simulated code inspection. Upon completion of the simulation, analysts were given a questionnaire requesting them to give their reactions as well as recommendations for the course. This questionnaire was followed by an interview with SwRI analysts. After completing the questionnaire and interview, the analysts were administered Posttest 2 (see Appendix A).

3.0 FINDINGS

3.1 Results of Data Collection

SwRI validated the relevance of this course to NASA's training needs by examining the course, comparing the course with the existing NASA code inspection standards/practice and evaluating the course's instructional effectiveness. This validation included an instructional design review of the subject matter (SwRI), subject matter expert review (SwRI and IBM), and an instructional effectiveness review (SwRI), including target audience trials (NASA and IBM analysts).

3.1.1 Content Review Findings

The following paragraphs describe the findings from the review of objectives, course content and underlying process models for code inspection.

3.1.1.1 Content Review by SwRI

SwRI compared the course content with the stated objectives from Carnegie Mellon. SwRI verified that the course content was consistent with the stated course objectives.

SwRI compared code inspection objectives as outlined in two NASA documents (Software Formal Inspections Guidebook and NASA Software Inspection Process Standard) with the objectives of the course. In order to make comparisons consistent in scope and depth, SwRI extracted more detail from the course and annotated the Carnegie Mellon objectives. SwRI found the annotated course objectives to be very similar to the NASA code inspection guidelines and standards.

3.1.1.2 Content Review by IBM

At the request of SwRI and NASA, analysts from IBM reviewed the annotated course objectives. IBM experts evaluated the detailed outline with respect to established NASA code inspection practice and procedures. SwRI also demonstrated the course to IBM experts in order to enhance their analyses. The following summary lists discrepancies between the course and NASA's code inspection process. The complete report from IBM appears in Appendix F.

Areas where the course content conflicts:

- characteristics of an inspection meeting
- differences between inspection and walkthrough procedures
- role of recorder

Areas that require additional detail:

- purpose of formal inspections
- stages of the formal inspection process
- benefits of the formal inspection process
- role of planning and preparation in the inspection process (planning)
- role of planning and preparation in the inspection process (preparation)

- review roles assumed during inspection
- types of checklists
- basic rules for code inspections
- moderator role description
- checklists and forms use in the formal inspections process
- importance of inspection as an organizational approach to ensure process stability/improvement

3.1.1.3 Code Inspection Model Review by SwRI

SwRI also compared models for code inspection. Carnegie Mellon provided a description of the model the course uses. The two documents previously mentioned, Software Formal Inspections Guidebook and NASA Software Inspection Process Standard, present NASA's model. SwRI found the two models to be similar. Only minor differences were detected.

3.1.1.4 Code Inspection Model Review by IBM

IBM reviewed the code inspection models and found the inspection process steps adequate. IBM moved the sequence of one step, exit criteria. IBM added the following required steps:

- re-work
- post meeting errors
- collection of inspection meeting reports
- submission of summary data to database
- extraction of reports from database
- summary metric data
- FACI/CI summary data

IBM added additional details to the following role descriptions:

- manager
- producer
- moderator
- recorder

IBM added the following roles as other individuals involved in the NASA inspection process:

- librarian
- designer/tester
- independent tester
- consumer

3.1.2 Instructional Effectiveness Review Findings

The following paragraphs describe the findings from the target audience trials (NASA and IBM) and the instructional effectiveness review (SwRI). This section states results as collected by various instruments; therefore, information may overlap by design (e.g., analyst questionnaire and interview questions). Raw data can be found in Appendix B.

3.1.2.1 Pretest/Posttest 1 (Knowledge of Information)

Analyst scores on the pretest (administered prior to using the course) were compared with their scores on Posttest 1 (administered after using the instructional modules). The purpose of this comparison was to determine how much analysts learned from the instructional modules (Knowledge of Information).

There was a definite improvement in ability to define terms after completing the instructional modules. When asked to answer questions on the pretest, such as stating the purposes of formal inspections or listing the stages of the formal code inspection process, analysts had difficulty producing completely correct answers. For example, on the pretest none of the analysts were able to list all stages in the formal code inspection process, whereas on Posttest 1 all analysts were able to list most, if not all, the stages. Multiple choice scores for all three analysts improved substantially from the pretest to Posttest 1 (see table below). Results for the multiple choice part of the test were consistent with results for the definition and fill in the blank parts of the test. Gains in scores from the Pretest to Posttest 1 indicate a substantial increase in knowledge of information about code inspection after completion of the course instructional modules.

Part 3 (Multiple Choice)	Analyst 1	Analyst 2	Analyst 3
Pretest	70%	74%	78%
Posttest 1	88%	86%	83%
Posttest 2	89%	87%	87%

Note: Scores indicate the percent correct out of a total 156 questions.

3.1.2.2 Posttest 1/Posttest 2 (Application of Information)

Analysts scores on Posttest 1 (administered after the instructional modules) were compared with their scores on Posttest 2 (administered after the simulation). The purpose of this comparison was to measure gain in knowledge as a result of application (did the analysts learn additional information from the simulation).

Changes in response from Posttest 1 to Posttest 2 on the definition part of the test were not substantial. Little additional information, if any, was noted, nor was there any noticeable improvement in quality of response. Similar results held true for the fill in the blank part of the test. Once again, very little additional information was noted on Posttest 2. No noticeable improvement in the quality of response was detected. Scores on the multiple choice part of the test increased slightly from Posttest 1 to Posttest

2 (see table above). Results for the definition, fill in the blank, and multiple choice parts of the test were consistent. Gains in scores were small, which indicates the analysts probably did not gain significant knowledge of information as a result of participating in the simulation.

3.1.2.3 Program Feedback

The course provides some feedback upon completion of a simulated code inspection. Strengths and weaknesses of an analyst's performance are given. A summary of the three analysts feedback is provided below. A complete listing of feedback for each analyst can be found in Appendix B.

A strength noted for two analysts was good use of emotional tone. All three analysts received praise for never introducing irrelevant topics, while two analysts were complimented for stopping tangents. Another analyst was praised for expressing a minority opinion and overall good participation.

The one weakness cited for all three analysts was missing two of the biggest errors in code. In addition, two analysts changed their opinions incorrectly (possibly due to group pressure). A weakness of one analyst was lack of input or being too passive during the inspection, while another was cited as being too aggressive. Another analyst had difficulty with the talk interface and left too many topics open without a final resolution.

3.1.2.4 Analyst Self Report

Analysts' reactions to the course were measured with an analyst questionnaire and an interview. Findings from these two instruments are presented below.

3.1.2.4.1 Analyst Questionnaire

The analyst questionnaire was divided into five sections. Results are summarized by section. A complete listing of results is found in Appendix B.

Overall Evaluation of the Course:

Analysts had a very positive attitude toward the course. They liked this method of instruction and preferred it over the way information is currently being taught. Analysts thought the course had numerous strengths, as well as some weaknesses which could be overcome. Two analysts strongly agreed this code inspection course has potential for use at NASA. The third analyst, who disagreed, does not currently perform code inspections. All analysts agreed they would like to see more courses of this type offered by NASA.

Course Content:

Two analysts agreed the code inspection model used in the course was similar to what is used at NASA (one analyst did not respond). The content of the course was very important (relevant) to one analyst and not as much to the other analysts (this correlated with their level of involvement on a daily basis in the code inspection process).

There were mixed feelings about the purpose of the course. Two analysts strongly agreed the purpose was clear, whereas one disagreed strongly. Two analysts agreed the content was academically challenging, whereas one analyst had no opinion. Finally, all three analysts agreed the level of detail in the course was appropriate for preparing someone to participate in a code inspection.

Learning Effectiveness:

All three analysts agreed they learned from the code inspection course. All also agreed they learned more from this method of instruction on code inspection than from current methods of instruction.

Instructional Presentation:

All analysts agreed the course captured their attention. Two analysts agreed the specific objectives of the course were not clear. In addition, prerequisite skills were not clear to all users. One analyst strongly agreed the course material was clear and well organized; the other analysts disagreed. One analyst thought there was enough practice provided, one was neutral, and a third did not respond. One analyst agreed strongly that feedback was adequate, one had no opinion and another did not respond. Two analysts agreed feedback was meaningful and one analyst did not respond. Analysts agreed the course's assessment of their performance was fair and meaningful.

All analysts strongly agreed that this method of instruction probably caused them to learn more than current methods for learning code inspection. All analysts strongly agreed they liked the method of instruction. Analysts also agreed the course was appropriate for their background and experience.

System Capabilities:

Two analysts agreed learning how to use the course was fairly easy, one strongly disagreed. Similarly, two analysts agreed learning how to use the simulation was fairly easy, one strongly disagreed. Similar reactions held true for remembering names and uses of commands. (Note: In the interview, analysts elaborated on how easy/difficult the course was to use and specified particular areas of difficulty.)

One analyst was frustrated during parts of the course, one did not have an opinion, and one was not too frustrated. Two analysts felt the simulation was slow. All analysts liked having a great deal of control over the instruction. Analysts agreed strongly the graphics were interesting and effective. They agreed the quality of the motion video was good and that it added value to the course.

3.1.2.4.2 Interview

The interview form was divided into the same five sections as the analyst questionnaire. The interview provided analysts an opportunity to elaborate on their responses on the questionnaire, as well as answer additional questions. Again, results are summarized by section. A complete listing of results is found in Appendix B.

Overall Evaluation of the Course:

Analysts liked this method of instruction. They all indicated they liked the full motion video scenarios best. Other strengths included: method of instruction (you remember the content longer), the simulation, and the library. Analysts indicated they disliked the following items: too much text on the screen, audio interferes with the text in places, the user interface (only parts, e.g. the mouse), and the section on groups in Module 5. Other weaknesses of the course included: lots of material to cover in one day, no instructions on how to operate the course, lack of instruction on how to use the tools, and not enough review provided. Analysts felt the weaknesses could be overcome. All analysts believed the course has potential for use in teaching the code inspection process.

Analysts recommended modifications to make the course more effective and easier to use. Specific applications for using this course were as training for new hires, as a review for experienced analysts, or in a workstation available for reference at any time. In addition, analysts recommended that instruction similar to this course (method of instruction) be used by NASA to teach other content as well.

Course Content:

One analyst felt the content of this course was particularly relevant at the present time, whereas the other two had no immediate need. All considered roles, behavior guidelines, and interpersonal communication skills the most relevant parts of the course. All analysts considered the segment on family and social groups to be irrelevant. Additional information desired included a segment on active listening, more video examples, more detail on some topics, easy access to definitions for unfamiliar terms and summaries at the end of sections. If called upon to use the course as is, analysts would use parts of the course which relate to the specific application at hand and omit the part on family and social groups. If the course were used, analysts recommend using the course over a period of time, rather than all in one day.

Learning Effectiveness:

All analysts said they learned from the course. They indicated they learned about the code inspection process, roles of participants, and what it is like to attend a real code inspection (scenarios and simulation). Two analysts indicated they learned general information, not details, because of the volume of information contained in the course and the limited time for using the course in this study.

Analysts agreed the content was appropriate for the course, but would make some changes. Analysts did emphasize the need for the course to clearly state a purpose and objectives.

Instructional Presentation:

Two analysts stated they liked the motion video segments in the program best. In addition, other desirable aspects of the course were the simulation and the library. Analysts indicated the following items were least liked about the method of presentation: too many text screens, audio sometimes competes with the text, lengthy introductions the user was required to sit through. The simulation, tools, and natural language interface were indicated as the most difficult parts of the program to use.

Analysts indicated they prefer this method of presentation for learning code inspection or this method in combination with current methods over the way code inspection is presently taught (OJT, manuals, working with experienced analysts). One analyst indicated the decision to use a course of this type would need to be weighed against cost.

System Capabilities:

Analysts indicated they had some difficulty with the natural language interface and how to use the tools. One analyst was particularly frustrated with the mouse and its placement on the screen. They did indicate that they felt these difficulties could be overcome.

3.1.2.5 Evaluator Observations

SwRI observed NASA and IBM analysts using the course. These observations support the opinions and recommendations expressed in both the questionnaire and the interview. A complete listing of observations can be found in Appendix B.

3.2 Summary of Findings

Following are the strengths and weaknesses of this course as summarized by SwRI after reviewing the course and using it with the NASA/IBM analysts.

3.2.1 Strengths of the Course

Following is a summary of the strengths of this course.

3.2.1.1 Instructional Issues

- Most of the content is easily understood.
- The level of difficulty of the content is appropriate for the target audience.
- Feedback on performance in the simulation is built into the program (simulation).
- The presentation of the content through full motion video is motivational (scenarios).
- Help is provided (although not context sensitive).
- The user controls pace of the instruction in almost all cases.
- The simulation helps the user apply what is learned.
- A library is provided for reference.
- The program provides a means for exiting at almost all times.

3.2.1.2 Aesthetic Issues

- In most cases, types of screens are consistent to provide navigation for the user (menu screens, text screens, etc.).
- Only one typographical error was found.

3.2.1.3 Technical Issues

- The course execution is consistent.

3.2.2 Weaknesses of the Course

Following is a summary of the weaknesses of this course.

3.2.2.1 Instructional Issues

- No purpose is stated for the course.
- Objectives are not clearly stated for the overall program or individual sections.
- Some content is missing and needs to be added.
- Some content needs to be changed.
- Some content seems irrelevant to the course (e.g., family and social groups).
- The model for code inspection used in the course is missing some steps.
- The order of steps in the model for code inspection used in this course needs to be changed.
- Directions for use of the course are not clearly stated.
- Complete documentation about the course needs to be provided (e.g., what it is, how to use it).

- The instructional sequence is not clearly stated.
- The instructional modules need to be more interactive.
- The user does not always have control over when to move to the next screen (not consistent).
- Better provision for reviewing sections needs to be included (some chunks of information could be smaller).
- The course needs to provide summaries/reviews at the end of each instructional module.
- The user needs to be able to pause and resume from that point.
- The user needs a provision to exit at most times (especially during lengthy introductions).
- No practice questions with feedback are given during the instructional modules.
- Evaluation criteria for the simulation is not explained.
- Help is not context sensitive.
- The text competes with audio in places and is not used consistently.
- The course needs better instruction on how to use the tools.
- The natural language interface is difficult to use and little instruction is provided.

3.2.2.2 Aesthetic Issues

- Screens are often packed with too much text.
- Color is not used effectively (text and background colors do not complement each other).
- Sections of motion video (e.g., motion video in the simulation that is not full motion) are unnatural (better than still frame, but not full motion).
- The mouse does not appear in a consistent location on the screen.

3.2.2.3 Technical Issues

- There are a few bugs in running the program; however, for the most part these are stated as limitations of the program (it must also be recognized that this is a prototype program).
- The program is slow in some places, especially the long simulation.
- The quality of the video is not clear during the simulation (partial motion video is lower quality compared with full motion video capability).
- The quality of the audio is poor in some places (e.g., simulation).

4.0 CONCLUSIONS

4.1 Conclusions of the Study

This study found the course, "A Cure for the Common Code", effective for teaching the process of code inspection. In addition, analysts prefer learning with this method of instruction or this method in combination with current methods. The unmodified course is definitely considered better than no course at all; however, findings indicate changes are needed. Our conclusions regarding the content, instructional effectiveness, and analysts' opinions about the course are presented below with a brief explanation for each.

4.1.1 Content

- Conclusion:** Some content needs to be added to the course.
Explanation: SwRI concluded that the scope of the course is adequate; however, there needs to be more depth in some areas. IBM liked the course but stated some items definitely need to be added to the course to reflect NASA's code inspection process (see Appendix F).
- Conclusion:** Some course content needs to be changed.
Explanation: In their review of the course objectives, IBM indicated that some information needed to be changed to customize the course to fit the NASA code inspection process (see Appendix F).
- Conclusion:** The steps in the code inspection process need to be more complete to closely follow the NASA model.
Explanation: In their review of the code inspection model, IBM indicated that some steps were missing from the Carnegie Mellon model and would need to be added to more accurately reflect the NASA model for code inspection (see Appendix F).
- Conclusion:** The order of the steps in the Carnegie Mellon code inspection model need to be changed to more closely follow the NASA model.
Explanation: In their review of the code inspection model, IBM indicated the order of steps did not accurately reflect the NASA model (see Appendix F).

4.1.2 Instructional Effectiveness

- Conclusion:** The course is instructionally effective.
Explanation: Based on gain scores between the pretest and Posttest 1, the course demonstrated an ability to teach the stated objectives.
- Conclusion:** The simulation has a positive effect on students' confidence in their ability to apply new knowledge.
Explanation: Although there was no meaningful gain in scores between Posttest 1 and Posttest 2 (no gain in knowledge of information as a result of application), analyst comments strongly indicated they favor the opportunity to practice code inspection and receive feedback on their performance while still in the training

environment. They strongly agreed with the concept but did not think this particular simulation was as effective as it could be.

Conclusion: Program feedback indicates that subjects were able to apply their new knowledge.
Explanation: All three analysts received positive feedback from the course on strengths they exhibited in the simulation. In addition, weaknesses were also presented. The weaknesses the course detected are consistent with limitations of the study and of the course tools. See Appendix B for specific program feedback.

4.1.3 Analyst Opinions

Conclusion: Analysts like the course and prefer this method of training, or this method in combination with current methods of training in code inspection, over the way training is currently being conducted.

Explanation: Overall, analysts appreciated this method of training (incorporating simulation, full motion video, scenarios), although they had reservations about many of the specifics of this particular course. These favorable reactions can be seen in both the analyst questionnaire and the interview responses.

Conclusion: Analysts responded favorably to information presented through scenarios incorporating full motion video.

Explanation: Analysts indicated throughout the course a desire to see more scenarios. This recommendation was emphasized again in the questionnaire and the interview.

5.0 LIMITATIONS

This study had a number of limitations which may have affected results. These limitations are summarized below.

5.1 Limitations of the Study

- The small sample size precluded a formal statistical analysis of the data.
- The pretest and the two posttests were exactly the same tests. To create different versions of tests that measure the same knowledge would have required considerably more resources. While retesting with the same test is a reliable way to measure a gain in knowledge, some of the gain on later scores may be attributed to a familiarity with the test. Also, a pretest can enhance learning by serving as an advance organizer for topics that the student should pay more attention to.
- Each subject was required to complete the course trial in one day, instead of the way the course was originally designed. This may have negatively affected their performance due to fatigue. Ideally, students would only take a few lessons at a time and not sit through six hours of instruction, simulation, and testing in one day.
- Due to time constraints during the course trials, minimal time was provided to examine the sample code prior to the simulation. This limitation could cause weaker performance in finding errors in the code during the simulation.
- The subjects were not proficient in the Ada programming language. This limitation could cause weaker performance in finding errors in the code during the simulation.
- For two subjects, code inspection was not part of their current job. This may have negatively affected their motivational interest in the course topic.

5.2 Limitations of the Course

- A "Cure for the Common Code" is a prototype course, not a polished product intended for distribution. The student instructions and supporting documentation are very sparse.
- The DVI hardware (7 board set) used to develop and deliver the course is outdated.
- The options in the natural language interface are limiting. The user may not be able to construct the exact response desired from the options provided by the natural language interface.
- The audio quality is lower than it could be with this technology.
- Some of the simulation environment does not utilize full motion video. The sequenced still frame displays look unnatural and jerky. This method, however, is probably more effective than just displaying one still frame.
- The interlaced monitor mode can result in eye fatigue.

5.3 Limitations of the Course as cited by Carnegie Mellon

5.3.1 General

- All inspection forms are not implemented in the program.
- There are intermittent problems with some CD-ROM drives ("Critical Error Occurred").

5.3.2 Training Room

- Early exits are not available from the orientation session within the training room and from the tool descriptions within Module 4.
- The student cannot exit the practice inspection and return later to the same state. This feature is available for the actual simulation.

5.3.3 Library

- The text materials in the library are incomplete.
- The user is forced to sit through the orientation session in the library during the first visit.

5.3.4 Conference Room

- The instructions for using the talk interface are minimal.
- The quality of the audio in parts of the simulation is poor (DVI configuration problems).
- The rule base is incomplete, so occasionally the participant will say something that is logical but makes no sense to the system.
- The audio is not well synchronized to the video (DVI 2.12 limitations).
- With the large inspection of the "procedure Options" code, there is a significant delay in audio file access from the CD-ROM. This increases as you progress with the inspection.

6.0 RECOMMENDATIONS

6.1 Conclusion Summary

Overall, this program appears to be effective for teaching the process of code inspection. In addition, analysts prefer learning with this method, or this method in combination with current methods.

6.2 NASA Options

Based on our conclusions, SwRI has outlined seven options for NASA to consider. The seven options are presented in the table below along with pros and cons for each.

OPTIONS	PROS	CONS	ESTIMATED COST TO IMPLEMENT
1. Take the Carnegie Mellon Course and use it as is (DVI 7 board set).	<ul style="list-style-type: none">• no new software development• this course is better than nothing	<ul style="list-style-type: none">• DVI 7 board set is unavailable• old technology (DVI version)• as is, the simulation is cumbersome to use• no technical support for the existing software (Carnegie Mellon)• no technical support for this version of DVI hardware or software (Intel)	none
2. Take the Carnegie Mellon Course and modify the existing course (DVI 7 board set).	<ul style="list-style-type: none">• could make minor changes to the data files of the course, not the source code (e.g., images, color, enlarge "hot spots") (non-instructional changes)	<ul style="list-style-type: none">• DVI 7 board set is unavailable• old technology (DVI version)• as is, the simulation is cumbersome to use• no technical support for consultation (Carnegie Mellon)• no technical support for this version of DVI hardware or software (Intel)	\$25,000

<p>3. Port and upgrade the Carnegie Mellon Course to a new system and modify the existing course as done in Option 2 (current DVI hardware).</p>	<ul style="list-style-type: none"> • new technology (higher quality version of interactive video) • could make minor changes to the data files of the course, not the source code (e.g., images, color, enlarge "hot spots") (non-instructional changes) 	<ul style="list-style-type: none"> • software development effort (porting and upgrading) • no technical support from Carnegie Mellon to port • limited technical support from Intel (regarding the old version) • as is, the simulation is cumbersome to use • possible compatibility problems with the existing expert module and new DVI software if the expert module is simply ported 	<p>\$75,000</p>
<p>4. Port and upgrade the Carnegie Mellon Course to a new system and modify the course design (current DVI hardware).</p>	<ul style="list-style-type: none"> • new technology (higher quality version of interactive video) • improve quality of instructional design (e.g., more scenarios) • add/change content • could replace the expert module with a better one 	<ul style="list-style-type: none"> • software development effort (porting, upgrading, and modifying) • no technical support from Carnegie Mellon to port • limited technical support from Intel (regarding the old version) • possible compatibility problems with existing expert module and new DVI software or software development effort for new expert module 	<p>\$100,000</p>

<p>5A Create new source code and data files using this course as a model (current DVI hardware).</p> <p>Note: This option would incorporate full motion video scenarios in place of a simulation.</p>	<ul style="list-style-type: none"> • new technology (higher quality version of interactive video) • improve quality of instructional design and tailor to NASA process • add/change content • not dependent on Carnegie Mellon for support • use lessons learned from the existing course • a portion of the instructional development is already done • a simulation would not have to be created (including the natural language interface) 	<ul style="list-style-type: none"> • major software development effort • scenario development effort 	<p>\$300,000</p>
<p>5B Create new source code and data files using this course as a model (current DVI hardware).</p> <p>Note: This option involves creation of a new simulation (including the natural language interface).</p>	<ul style="list-style-type: none"> • new technology (higher quality version of interactive video) • improve quality of instructional design and tailor to NASA process • add/change content • not dependent on Carnegie Mellon for support • use lessons learned from the existing course • a portion of the instructional development is already done 	<ul style="list-style-type: none"> • major software development effort • simulation development effort (including the natural language interface) 	<p>\$400,000</p>

6. Create a totally new course (current DVI hardware).	<ul style="list-style-type: none"> • new technology (higher quality version of interactive video) • improve quality of instructional design and tailor to NASA process • not dependent on Carnegie Mellon for support • use lessons learned from existing course 	<ul style="list-style-type: none"> • most expensive option • extensive software development effort • more instructional design effort involved 	\$500,000
--	--	---	-----------

6.3 SwRI Recommendation

SwRI believes this study indicates interactive video instruction combined with simulation is effective for teaching software processes. SwRI believes either option Five A or Five B will produce the most effective results. Options Five A and Five B are the same with the exception of the simulation. Both options involve creation of new source code and data files, but use much of the existing content and course design. Although both options involve a significant software development effort, many benefits are gained. Both options incorporate new technology which will produce higher quality audio and video. Content can be changed and added, and the quality of instructional design can be improved to tailor the course to the NASA process. The instructional development effort is minimized by modeling the existing course. In addition, lessons learned from the existing course can be applied to the new course. Finally, by creating a new course, NASA is not dependent on Carnegie Mellon for support.

Option Five B includes creation of a new full simulation. SwRI recognizes that creation of a full simulation is expensive; therefore an alternative is offered in option Five A which will provide many of the benefits of a full simulation at a lower cost. A major strength of the simulation in the existing course is that it gives the user scenarios to learn from. The alternative option, Five A, gives the user access to the data base of the expert system; however, it becomes menu driven, making it easy for a user to access specific information desired. By implementing this alternative option (Five A), creation of a new natural language interface needed for a full simulation is also avoided. Following is a brief summary of option Five A:

- Rework the instructional modules incorporating modifications as recommended in the following section.
- Instead of creating a new simulation, create scenarios using full motion video. These scenarios could be incorporated into the instructional modules, or included as a separate part of the course.

Scenarios from the existing course could be expanded and new scenarios could be created, eliminating many of the text screens contained in the existing course. Scenarios could be developed to illustrate:

- each phase in the code inspection process
- roles on a code inspection team

- variables affecting a code inspection meeting (e.g., individual personalities, level of preparation)

6.3.1 Modification Suggestions

Both options Five A and Five B involve use of some content and design from the existing course. SwRI suggests the following modifications be made to existing parts used, in order to meet the needs of NASA analysts.

6.3.1.1 Content

- state the purpose of the course
- state the objectives of the course
- add content to the course per recommendations (see Appendix F)
- change content in the course per recommendations (see Appendix F)
- add steps in the model for code inspection per recommendations (see Appendix F)
- change order of steps in the model for code inspection per recommendations (see Appendix F)
- include more full motion video scenarios to present information
- organize content more carefully (e.g., it is confusing if the producer is discussed in the section on the moderator)
- omit the section on family and social groups and make the remaining content on groups relevant to code inspection
- provide on-line, context sensitive help

6.3.1.2 Presentation

- present content in smaller chunks within the instructional modules
- make the instructional modules more interactive (e.g., insert practice questions with feedback)
- provide summaries/reviews at the end of sections within the instructional modules
- place less text on each screen (more white space)
- choose text and background colors which make the instruction more readable (contrast between text and background)
- support text with audio (audio should not contradict or interfere with visuals)
- use audio consistently with each screen (or indicate there is no audio with a particular screen)

6.3.1.3 User Interface Features

- provide instructions on how to use the program (e.g., floorplan)
- provide instructions for navigating in the program (describe buttons or make them more descriptive of their action)
- provide a method for exiting the program during introductory segments (e.g., first time through instructional modules, library)
- give the user control over when to proceed to the next screen (consistent)
- make provisions for the learner to pause at any time and resume from that point
- place the mouse on the screen in the position where the user is most likely to click
- state the purpose of the tools and when they are available to the user
- provide more and better instruction on how to use the tools

- provide more and better instruction on how to use the natural language interface in the simulation
- make the simulation respond more quickly

In summary, SwRI recommends option Five A as the most instructionally effective and the most cost effective option for incorporating process simulation training into current training efforts.

7.0 REFERENCES

NASA (August, 1991). Software Formal Inspections Guidebook.

NASA (December 9, 1991). NASA Software Inspection Process Standard.

Software Engineering Institute, Carnegie Mellon University (February, 1991). Instructions for ALT Course Administrators.

Software Engineering Institute, Carnegie Mellon University (February, 1991). Instructions for ALT Code Inspection Course Users.

APPENDIX A
INSTRUMENTS
(PRETEST/POSTTEST)

Name: _____

Start Time: _____ Completion Time: _____

CODE INSPECTION COURSE

- Posttest -

Part 1: Definitions

Directions: Define each of the following terms.

1. Define the following roles.
 - A. Moderator:
 - B. Reader:
 - C. Recorder:
 - D. Producer:
2. Define a formal software inspection.
3. Define a code walkthrough.
4. Define a formal design review board.
5. Define a task-oriented group.

Name: _____

Start Time: _____

Completion Time: _____

CODE INSPECTION COURSE

- Posttest -

Part 2: Fill in the Blank

Directions: Answer each of the following questions.

1. List the purposes of formal inspections.
2. List, in order, the stages of the formal inspection process.
3. List the benefits of the formal inspection process.
4. List the roles that participants assume during a code inspection.
5. List the types of checklists that can be used before or during the code inspection.
6. List characteristics of a code inspection meeting such as length and role responsibilities.
7. List basic rules for code inspection such as constraints regarding time, roles, or purposes.

8. List behavioral guidelines (for participants) that help code inspections succeed.
9. List the functions of formal software inspections.
10. List the functions of code walkthroughs.
11. List the functions of formal design review boards.
12. List advantages and disadvantages of formal software inspections.
13. List advantages and disadvantages of code walkthroughs.
14. List advantages and disadvantages of formal design review boards.
15. List techniques other than formal software inspections, code walkthroughs and formal design review boards for assuring software quality.
16. List problems often faced by moderators in conducting a software inspection.
17. List some potential problem situations emerging from interaction within the group during inspection.

18. List some report forms used in the formal code inspection process.
19. List common problems within a task-oriented group.
20. List characteristics of successful task-oriented groups.

Name: _____

Start Time: _____ Completion Time: _____

CODE INSPECTION COURSE

- Posttest -

Part 3: Multiple Choice

Directions: Circle all correct answers. For each question there may be more than one correct choice.

1. Which of the following purposes apply to formal code inspections?
 - A. to promote adherence to project style and rules of construction
 - B. to promote compliance with technology practices
 - C. to obtain metrics on the code producer's performance
 - D. to obtain metrics for project management and process control

2. Which of the following formal stages are a part of the code inspection process?
 - A. planning
 - B. writing the code
 - C. reinspection
 - D. preparation

3. Which of the following benefits are a result of the formal code inspection process?
 - A. improves error detection
 - B. integrates developer, user, and customer feedback
 - C. improves productivity
 - D. selects solutions to software errors

4. Which of the following tasks occur during the planning stage?
 - A. distribute inspection packages to participants
 - B. select the moderator
 - C. select inspectors and assign roles
 - D. schedule inspection meetings

5. Which of the following apply to the overview stage?
- A. often led by the producer
 - B. confirm schedule and receipt of materials
 - C. education on code inspection
 - D. background information given on work product
6. Which of the following are part of the preparation stage?
- A. verifies workproduct meets entry requirements
 - B. participants locate possible defects
 - C. participants gain knowledge of workproduct
 - D. participants brainstorm possible solutions for defects
7. Do managers assume a review role during the code inspection?
- A. yes
 - B. no
8. Which of the following may be used as a checklist either before or during the code inspection?
- A. construction rules
 - B. style guides
 - C. test cases
 - D. metrics checklist
9. Which of the following describe(s) a formal code inspection?
- A. small peer group process
 - B. purpose is detection and correction of software product defects
 - C. external review process
 - D. rigorous entry and exit criteria
10. Which of the following basic rules apply to code inspection?
- A. management should not be present at inspections
 - B. inspections are a tool for worker evaluation
 - C. producers should not be the moderator of their own work
 - D. inspections should be limited to approximately 2 hours

11. Which of the following guidelines apply to successful code inspections?
- A. have at least one positive comment
 - B. record all issues in public
 - C. stick to technical issues
12. Which of the following involves an external group examining the product?
- A. software inspection
 - B. code walkthrough
 - C. formal design review board
13. Which of the following describe(s) a code walkthrough?
- A. may be informal or structured
 - B. method for early defect detection
 - C. external process review
 - D. may be large or small peer groups
14. Which of the following could be a potential disadvantage to software inspections?
- A. focuses on producer's perspective
 - B. process stifles creativity
 - C. provides early quantitative quality evaluation
 - D. provides historical error database to reduce recurrences
15. Which of the following could be a potential disadvantage to code walkthroughs?
- A. the timing of defect detection
 - B. collective review of possible problems
 - C. varying structure yields inconsistent results
 - D. focuses on producer's perspective
16. Which of the following could be a potential disadvantage to formal design review boards?
- A. integrates the developer, user, and customer perspective
 - B. seldom challenges the technical basis of design
 - C. does not furnish management visibility for approval/disapproval of proceeding to next phase
 - D. focuses on producer's perspective

17. Which of the following individuals is responsible for initiating the inspection meeting?
- A. moderator
 - B. reader
 - C. manager
 - D. producer
18. In the planning stage, which individual verifies with the producer that the workproduct meets entry criteria?
- A. moderator
 - B. reader
 - C. recorder
 - D. manager
19. Which of the following individuals is responsible for compiling and recording preparation times from the preparation logs?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
20. In the planning stage, the producer must provide which of the following?
- A. function descriptions
 - B. comments
 - C. detailed design materials
 - D. support documentation
21. During the overview stage, who is the most active participant?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
22. During the overview, which individual must be able to paraphrase the workproduct for other members?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer

23. During the overview, which individual is responsible for answering detailed questions for the group regarding the workproduct?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
24. During the inspection meeting, which individual is responsible for determining preparedness to continue?
- A. moderator
 - B. reader
 - C. manager
 - D. producer
25. During the inspection meeting, which individual introduces the team and states the purpose of the meeting?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
26. During the inspection meeting, which individual determines the disposition of the workproduct?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
27. During the inspection meeting, which individual paces the group?
- A. manager
 - B. reader
 - C. recorder
 - D. producer

28. During the inspection meeting, which individual notes location, description, class, and type of defect?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
29. Which individual is ultimately responsible for keeping the meeting around the designated length of time and for closing the meeting?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
30. During rework, which individual verifies that defect corrections are made?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
31. During rework, which individual is responsible for correcting defects listed on the Inspections Defect List?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
32. During followup, which individual is responsible for completing the inspection management report?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer

33. During followup, which individual is responsible for consulting with the moderator to verify that corrections have been completed?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
34. Which individual is responsible for scheduling a reinspection, if necessary?
- A. moderator
 - B. reader
 - C. recorder
 - D. producer
35. The purpose of the preparation log is:
- A. to record how long it took the producer to write the code to be inspected
 - B. to record how long each participant took to prepare for the inspection
 - C. to record how long the inspection meeting lasted
 - D. to record how long the moderator spent preparing for the inspection meeting
36. The purpose of the inspection defect list is:
- A. to provide a record of points brought up during the inspection
 - B. to provide a record of preparation done for the inspection
 - C. to identify solutions
 - D. to provide statistics about the producer's performance
37. The purpose of the code inspection summary report is:
- A. to provide a summary of the producer's performance
 - B. to provide a record of how long the inspection meeting lasted
 - C. to provide a summary of each individual's performance in the inspection meeting
 - D. to provide a compilation of defects passed on to the moderator
38. The purpose of the management summary report is:
- A. a detailed report incorporating information from the code inspection summary report
 - B. a report of the inspection meeting by the producer's manager
 - C. a report to management regarding the producer's level of work
 - D. to provide a list of defects to management for review

39. Which of the following is not a sign of a good inspection?
- A. accurate assessment of the workproduct
 - B. defects detected efficiently
 - C. solutions arrived at for defects found
 - D. cooperation between group members
40. Which of the following is not characteristic of a task-oriented group?
- A. members are actively involved in group problem solving
 - B. comes together to accomplish goal or task
 - C. achieves goals through effective communication
 - D. organized in an informal way

APPENDIX A
INSTRUMENTS
(ANALYST QUESTIONNAIRE)

Name: _____

Date: _____

Start Time: _____

Completion Time: _____

ANALYST QUESTIONNAIRE

Directions: Circle the number corresponding to your opinion about the statement. Add comments where appropriate.

STATEMENT

RATING

Strongly
Disagree

No
Opinion

Strongly
Agree

Overall Evaluation of the Course

- | | | | | | |
|--|---|---|---|---|---|
| 1. I liked this method of instruction. | 1 | 2 | 3 | 4 | 5 |
| 2. I prefer this method of instruction to the way information is currently being taught. | 1 | 2 | 3 | 4 | 5 |
| 3. This course was motivational and held my interest. | 1 | 2 | 3 | 4 | 5 |
| 4. I think this course has a number of strengths that make it appealing. | 1 | 2 | 3 | 4 | 5 |
| 5. I think this course has some weaknesses that need to be overcome. | 1 | 2 | 3 | 4 | 5 |
| 6. The course was too long and involved for me. | 1 | 2 | 3 | 4 | 5 |
| 7. I think this specific course has potential for use at NASA. | 1 | 2 | 3 | 4 | 5 |
| 8. I can apply the skills I have learned in this course to my job. | 1 | 2 | 3 | 4 | 5 |
| 9. I would like to see more courses of this type offered by NASA. | 1 | 2 | 3 | 4 | 5 |

	Strongly Disagree		No Opinion		Strongly Agree
Course Content					
1. The code inspection model used in this course is similar to what I currently use.	1	2	3	4	5
2. The content of this course is important / relevant to me.	1	2	3	4	5
3. The purpose of this course was clear to me.	1	2	3	4	5
4. The course content was academically challenging for me.	1	2	3	4	5
5. The level of detail in this course was appropriate for preparing me to participate in a code inspection.	1	2	3	4	5
Learning Effectiveness					
1. I learned about code inspection from this course.	1	2	3	4	5
2. I learned more from this method of instruction than other current methods of instruction.	1	2	3	4	5
Instructional Presentation					
1. The course captured my attention.	1	2	3	4	5
2. I understood what the objectives of this course were.	1	2	3	4	5
3. Prerequisite skills for this course were made clear to the user.	1	2	3	4	5
4. The course material was clear and well organized.	1	2	3	4	5
5. Overall, enough opportunity was given for me to practice what I learned.	1	2	3	4	5
6. Course feedback to me was adequate.	1	2	3	4	5

	Strongly Disagree		No Opinion		Strongly Agree
7. Course feedback to me was meaningful.	1	2	3	4	5
8. Assessment of my performance was fair.	1	2	3	4	5
9. Assessment of my performance was meaningful.	1	2	3	4	5
10. Because of the method of presentation (multimedia, interaction, simulation, etc.), I believe I learned more than with current methods for learning code inspection.	1	2	3	4	5
11. Overall, I liked the method of presentation (multimedia, interaction, simulation, etc.) used in this course.	1	2	3	4	5
12. This type of course was appropriate for my background and experience.	1	2	3	4	5

System Capabilities

1. Overall, learning how to use the course was easy for me.	1	2	3	4	5
2. Specifically, learning how to use the simulation was easy for me.	1	2	3	4	5
3. Remembering names and uses of commands was easy for me.	1	2	3	4	5
4. I was frustrated during parts of the course.	1	2	3	4	5
5. The course speed / response was too slow / cumbersome.	1	2	3	4	5
6. Letting me control where I went added value to the instruction.	1	2	3	4	5
7. Graphics were interesting and effective.	1	2	3	4	5
8. I could understand the audio well.	1	2	3	4	5

	Strongly Disagree		No Opinion		Strongly Agree
9. The quality of the video (full motion) was good.	1	2	3	4	5
10. The video (full motion) added value to the course.	1	2	3	4	5

APPENDIX A
INSTRUMENTS
(INTERVIEW QUESTIONS)

Name: _____

Date: _____

Start Time: _____

Completion Time: _____

INTERVIEW QUESTIONS

Overall Evaluation of the Course

1. Did you like this method of instruction? Why or why not?
2. Overall, what did you like most about this course?
3. Overall, what did you like least about this course?
4. What do you think the strengths of this course are?
5. What do you think the weaknesses of this course are?

Can these weaknesses be overcome?

What are some suggestions for overcoming these weaknesses?

6. Does this course have potential for use in teaching the code inspection course?
7. Would you recommend this specific course be used at NASA/IBM for training analysts in the code inspection process? Is it better than nothing?

What changes would you recommend?

Why would you make these changes?

8. Would you recommend instruction similar to this course be used by NASA/IBM to teach other content or processes?

9. What do you think would improve this course the most?

10. How would you suggest using this course?

Learning Effectiveness

1. Do you feel you learned from this course? Why or why not?

2. If yes, what specifically did you learn?

3. What changes would make this course more effective in teaching the content?

Instructional Presentation

1. What did you like best about the method of presentation of this course?

2. What did you like the least about the method of presentation of this course?

3. What parts were difficult to use? Why? Be specific.

4. How does this course compare to how you currently receive instruction in the code inspection process?

Which method do you prefer for learning?

Course Content

1. Is the content of this course relevant to you?

2. What content specifically is most relevant for you?

What content specifically is most irrelevant for you?

What content is missing that you view as relevant and should be added to the course?

3. If you used this course as it presently exists, what parts would you use in terms of content?

What parts, if any, would you omit in terms of content?

Other general suggestions for use?

4. How could this course content be changed to more closely fit NASA needs?
5. What other content/processes do you think this type of instruction might be appropriate for? Be specific.
6. How realistic were the video scenarios during the instructional modules?
7. How realistic were the video scenarios during the simulation?

System Capabilities

1. Did you have difficulties with any sections of the course?

If yes, what areas of the course did you have difficulty with?

Do you think these difficulties could be overcome?

What suggestions do you have for overcoming these difficulties?

2. What is your opinion of the ease of use in the following parts of the course?

Auditorium:

Training Room:

Module 1 (Formal Inspections: Purpose and Process):

Module 2 (Inspections Types and Differences):

Module 3 (Inspection Roles and Pitfalls):

Module 4 (Inspections Tools and Forms):

Module 5 (Inspections Communications):

Library:

Office:

Conference Room (simulated code inspections):

APPENDIX A
INSTRUMENTS
(OBSERVATION FORM)

Name: _____

Date: _____

Start Time: _____

Completion Time: _____

OBSERVATION FORM

TASK	PROBLEMS/ DIFFICULTIES	ANALYST COMMENTS	OBSERVER COMMENTS	TIME SPENT
------	---------------------------	---------------------	----------------------	---------------

Code Inspection Course
Auditorium

Training Room
Module 1

Module 2

Module 3

Module 4

Module 5

Posttest (written)

Code Inspection Course
Practice Inspection

Library

Office

Code Inspection

Analyst Questionnaire

Interview Questions

Posttest (written)

APPENDIX A
INSTRUMENTS
(DEMOGRAPHIC DATA SHEET)

DEMOGRAPHIC INFORMATION

Name: _____

Title: _____

Degree(s): _____

Computer Experience

1. How many years have you used computers?
2. How many hours per day do you currently use computers in your work?
3. For what applications do you currently use computers (programming, word processing, etc.)?

Programming Experience

1. How many college level courses have you taken where you were required to write or understand a program in a procedural programming language such as Pascal, FORTRAN, C, Ada, etc?
2. How many college-level courses have you taken where you were required to write or understand a program in the Ada programming language? Briefly describe your level of experience with the Ada programming language.
3. How many computer languages can you understand and program with? Please list these languages.

Code Inspection Process

1. Have you ever participated in a code inspection before? If yes, how many?
2. How long have you been involved in the code inspection process?
3. Is code inspection part of your present job?

4. When was the last time you participated in a code inspection? What roles did you perform?
5. Have you ever received formal training in the code inspection process? If yes, please describe briefly.
6. Have you taken any college level courses where software engineering concepts were taught? If yes, how many? Please describe briefly.
7. Have you taken any college level courses where software technical reviews were discussed? If yes, how many? Please describe briefly.

General Information

1. Have you ever received training via the computer before (computer-based training)?

If yes, how many courses and for what topics?

Do you feel like it was an effective way to learn?

Did you like learning from computer-based training?

What do you feel are some strengths of computer-based training?

What do you feel are some weaknesses of computer-based training?

2. Have you ever received any training which incorporates audio and video images? If yes, please describe.
3. Have you ever received any training which incorporates a process simulation, for example code inspection, before? If yes, what process was taught? Please describe briefly.

APPENDIX A
INSTRUMENTS
(INSTRUCTIONAL PATH)

Name: _____ Date: _____

Start Time: _____ Completion Time: _____

NASA CODE INSPECTION INSTRUCTIONAL VALIDATION

_____ **Demographic Data Sheet**

_____ **Pretest (written)**

_____ **Code Inspection Course**

_____ **1. Auditorium (8 minutes)**

_____ **2. Training Room**

_____ **A. Log In**

_____ **B. Module 1 - Formal Inspections: Purpose and Process (14 minutes)**

_____ **C. Module 2 - Inspection Types and Differences (20 minutes)**

_____ **D. Module 3 - Inspection Roles and Pitfalls (43 minutes)**

_____ **E. Module 4 - Inspection Tools and Forms (14 minutes)**

_____ **F. Module 5 - Inspection Communications (29 minutes)**

_____ **Posttest (written)**

_____ **G. Practice Inspection - "Options" (choose moderator from office) (15 min.)**

_____ **Record Strengths/Weaknesses**

_____ **3. Library (10 minutes for orientation and initial exploration)**

_____ **4. Office (5 minutes for exploration)**

_____ **5. Training Room or Library (return occasionally)**

_____ **6. Conference Room (inspection of the "Find_Maximum" code) (recorder) (45 min.)**

_____ **Record Strengths/Weaknesses**

_____ **Analyst Questionnaire**

_____ **Interview Questions**

_____ **Posttest (written)**

NASA CODE INSPECTION INSTRUCTIONAL VALIDATION

1. "Options" (Practice Inspection)

Strengths:

Weaknesses:

2. "Find_Maximum" (Final Inspection)

Strengths:

Weaknesses:

APPENDIX B
RAW DATA
(PROGRAM FEEDBACK)

PROGRAM FEEDBACK

Analyst	Strengths	Weaknesses
1	<ul style="list-style-type: none"> • good use of emotional tone • never introduced irrelevant topics 	<ul style="list-style-type: none"> • missed the two biggest errors in the code • hanged opinion incorrectly, perhaps due to group pressure • lack of input from you (review inspection communication module) • too passive during inspection
2	<ul style="list-style-type: none"> • good use of emotional tone • never introduced irrelevant topics • good job stopping tangents as well 	<ul style="list-style-type: none"> • missed the two biggest errors in the code • changed opinions incorrectly, perhaps due to group pressure • difficulty with talk interface • too many topics left open without a stated final resolution
3	<ul style="list-style-type: none"> • never introduced irrelevant topics • good job stopping tangents • correctly expressed a minority opinion • good participation 	<ul style="list-style-type: none"> • missed the two biggest errors in the code • too many aggressive comments

APPENDIX B
RAW DATA
(ANALYST QUESTIONNAIRE)

Name: _____

Date: _____

Start Time: _____

Completion Time: _____

ANALYST QUESTIONNAIRE

Directions: Circle the number corresponding to your opinion about the statement. Add comments where appropriate.

STATEMENT

RATING

Strongly
Disagree
1 2

No
Opinion
3 4

Strongly
Agree
5

Analyst
#1

Analyst
#2

Analyst
#3

Overall Evaluation of the Course

- | | | | |
|--|---|---|---|
| 1. I liked this method of instruction. | 4 | 5 | 4 |
| 2. I prefer this method of instruction to the way information is currently being taught. | 5 | 5 | 4 |
| 3. This course was motivational and held my interest. | 5 | 4 | 4 |
| 4. I think this course has a number of strengths that make it appealing. | 5 | 4 | 5 |
| 5. I think this course has some weaknesses that need to be overcome. | 5 | 4 | 3 |
| 6. The course was too long and involved for me. | 4 | | 1 |
| 7. I think this specific course has potential for use at NASA. | 5 | 5 | 2 |
| 8. I can apply the skills I have learned in this course to my job. | 5 | 2 | 1 |

	Strongly Disagree 1	2	No Opinion 3	4	Strongly Agree 5
	Analyst #1		Analyst #2		Analyst #3
9. I would like to see more courses of this type offered by NASA.	5		5		4

Course Content

1. The code inspection model used in this course is similar to what I currently use.	4				3
2. The content of this course is important / relevant to me.	5		3		4
3. The purpose of this course was clear to me.	5		1		5
4. The course content was academically challenging for me.	4		3		4
5. The level of detail in this course was appropriate for preparing me to participate in a code inspection.	5		4		4

Learning Effectiveness

1. I learned about code inspection from this course.	5		5		4
2. I learned more from this method of instruction than other current methods of instruction.	5		5		4

Instructional Presentation

1. The course captured my attention.	4		5		4
2. I understood what the objectives of this course were.	5		1		2
3. Prerequisite skills for this course were made clear to the user.	4		1		3
4. The course material was clear and well organized.	5		2		2

	Strongly Disagree 1	2	No Opinion 3	4	Strongly Agree 5
	Analyst #1		Analyst #2		Analyst #3
5. Overall, enough opportunity was given for me to practice what I learned.	4				3
6. Course feedback to me was adequate.	5				3
7. Course feedback to me was meaningful.	4				4
8. Assessment of my performance was fair.	4				4
9. Assessment of my performance was meaningful.	4				4
10 Because of the method of presentation (multimedia, interaction, simulation, etc.), I believe I learned more than with current methods for learning code inspection.	5		5		5
11 Overall, I liked the method of presentation (multimedia, interaction, simulation, etc.) used in this course.	5		5		5
12 This type of course was appropriate for my background and experience.	5		4		4
System Capabilities					
1. Overall, learning how to use the course was easy for me.	4		2		4
2. Specifically, learning how to use the simulation was easy for me.	4		1		4
3. Remembering names and uses of commands was easy for me.	4		1		3
4. I was frustrated during parts of the course.	4		3		2
5. The course speed / response was too slow / cumbersome.	4		4		2
6. Letting me control where I went added value to the instruction.	5		4		4

	Strongly Disagree		No Opinion		Strongly Agree
	1	2	3	4	5
	Analyst #1		Analyst #2		Analyst #3
7. Graphics were interesting and effective.	5		5		4
8. I could understand the audio well.	5		5		4
9. The quality of the video (full motion) was good.	4		4		4
10 The video (full motion) added value to the course.	5		5		4

APPENDIX B
RAW DATA
(INTERVIEW QUESTIONS)

Name: _____

Date: _____

Start Time: _____

Completion Time: _____

INTERVIEW QUESTIONS

Overall Evaluation of the Course

1. Did you like this method of instruction? Why or why not?

Analyst 1: Yes
Helpful because of real examples (video)
Interesting
Allows you to practice (simulation)
Access to information in the library

Analyst 2: Yes
Really good, definitely worth it
Needs some fine tuning
Lots better than manuals
Available (come and get what need)
Variety
More interesting
Uses more senses (multimedia, many methods)

Analyst 3: Yes, better than manuals
Audio and video with words, get more out of it
Motion video sequence (what if's)

2. Overall, what did you like most about this course?

Analyst 1: Scenarios (video)
Analyst 2: Video examples (especially if interactive)
Analyst 3: Video segments

3. Overall, what did you like least about this course?

Analyst 1: Talk too much
Too much information on the screen
Audio interferes with text on screen

Analyst 2: User interface (mouse, consistency, prompt not clear) (should be obvious)

Analyst 3: All pretty good
Cursor moving around
Group section
Audio doesn't always follow text

4. What do you think the strengths of this course are?

Analyst 1: Help you to remember information for longer time because of way it is presented

Analyst 2: See #1
Analyst 3: Same as above

5. What do you think the weaknesses of this course are?

Analyst 1: Lots of material
No instructions on how to operate the course
Analyst 2: Group section not in code inspection context
Program is unclear if tools could be used outside of the course
Waste time learning about tools you only use for course, not in reality
Not being able to navigate to certain parts for review
Audio competes with text at times
User interface
Analyst 3: Same as above

Can these weaknesses be overcome?

Analyst 1: Yes
Analyst 2: Yes, definitely, no doubt
Analyst 3: Yes

What are some suggestions for overcoming these weaknesses?

Analyst 1: Break up material
Summary/review at end of modules
Be able to review smaller chunks of information
Instructions on how to operate the course
Analyst 2: Scenarios as opposed to lengthy text
Analyst 3: Separate course on groups and include more items on effective meetings (relate to code inspection)

6. Does this course have potential for use in teaching the code inspection course?

Analyst 1: Yes
Analyst 2: Yes, definitely
Analyst 3: Yes (introduction for a new hire, reference for more experienced employees, use to relieve first time tensions associated with code inspection)

7. Would you recommend this specific course be used at NASA/IBM for training analysts in the code inspection process? Is it better than nothing?

Analyst 1: No, needs modifications
Still better as is than no course at all
Analyst 2: Definitely better than nothing, can get something out of it
What matters is if the content is right for NASA
Analyst 3: Hard to say (don't know what they do over there)
Cost is a consideration

What changes would you recommend?

Analyst 1: Change weaknesses in #5

Analyst 2: Minimal
More video (helps retain information)
User interface
Some content presented in such a way that I didn't retain it
Content (better organization or structure of content, hand hold me better through it)

Analyst 3: Same as #2-5

Why would you make these changes?

Analyst 1: To make class more effective
To learn more

Analyst 2: Ease of use

Analyst 3:

8. Would you recommend instruction similar to this course be used by NASA/IBM to teach other content or processes?

Analyst 1: Sure

Analyst 2: Definitely

Analyst 3: Requirements inspections
Level six test case review

9. What do you think would improve this course the most?

Analyst 1: More scenarios, examples, and video

Analyst 2: User interface

Get to sections easily and just use parts

Analyst 3: More video segments

10. How would you suggest using this course?

Analyst 1:

Analyst 2: For new hires, experienced people for review, or workstation (reference)

Analyst 3: See #6

Learning Effectiveness

1. Do you feel you learned from this course? Why or why not?

Analyst 1: Yes

Analyst 2: Yes

Analyst 3: Yes, when took test, couldn't recall, but could recognize (multiple choice okay, still hard on parts 1 and 2)
Too much information in too little time

2. If yes, what specifically did you learn?

Analyst 1: Never knew certain things about code inspection
Roles of people

- Not only learned the material, but feel like I really attended a code inspection (scenarios and instructional modules both)
- Analyst 2: General information, not details (the information is there, but it is overwhelming and hard to get to). Important how you section information and present to people
- Analyst 3: Overall "process" of code inspection
Roles
3. What changes would make this course more effective in teaching the content?
- Analyst 1: Content is pretty good
- Analyst 2: Need a purpose
Need objectives
Provide a course description or objective for a person so they could use or not use (waste time)
- Analyst 3: See previous question

Instructional Presentation

1. What did you like best about the method of presentation of this course?
- Analyst 1: Video
Library to access information
- Analyst 2: Video examples (retention)
- Analyst 3: Simulation (experience of being in an inspection without actually being involved in one)
2. What did you like the least about the method of presentation of this course?
- Analyst 1: Coffee room & office were redundant
Don't need these rooms, can practice in training room
- Analyst 2: Text screens
Audio at times detracted from video
- Analyst 3: Sometimes introductions were too long (I am wasting my time listening to this person)
Doing the course all in one day was hard
No major complaints
3. What parts were difficult to use? Why? Be specific.
- Analyst 1: Simulation (in general)
Something to bring back to main menu so can proceed quickly
- Analyst 2: Tools
Natural language interface
- Analyst 3: None really, it was pretty simple to use
Tools and natural language interface in modules, program doesn't tell you they are just for this training and when they will be used
4. How does this course compare to how you currently receive instruction in the code inspection

process?

Analyst 1: Have never received one
On the job training

Analyst 2: Learned lots from this course
It is a lot better than manuals

Analyst 3: Suggest following this course up with a code inspection where you just observe
and then discussion with experienced person (the 3 go together)

Analyst 3: Better, but weigh decisions with cost of producing training for all the areas
It's either "on the job training" or this
Don't think print base works very well at all

Which method do you prefer for learning?

Analyst 1: This course first and then on the job training

Analyst 2: Use this program, observe a code inspection, discuss with experienced person

Analyst 3:

Course Content

1. Is the content of this course relevant to you?

Analyst 1: Yes

Analyst 2: No, I don't do code inspection but I did want to know

Analyst 3: No

2. What content specifically is most relevant for you?

Analyst 1: Role of reviewers

Behavior guidelines

Interpersonal communication skills

Analyst 2: Group dynamics

Analyst 3:

What content specifically is most irrelevant for you?

Analyst 1: Different kinds of groups (family, social)

Analyst 2: Family and social groups (most people know this). Doesn't relate to code
inspection (obvious, who cares)

Analyst 3:

What content is missing that you view as relevant and should be added to the course?

Analyst 1: Summary at the end of sections

Analyst 2: Active listening (use scenarios) is really important

More video examples (icon available if you want to see more videos)

More information/detail on some things

Better definitions of terms (on line glossary)

Order of content (not missing, but put in different location)

Analyst 3:

3. If you used this course as it presently exists, what parts would you use in terms of content?
Analyst 1: All except those listed in the next question
Analyst 2: Depends on application (use what need)
Analyst 3:

What parts, if any, would you omit in terms of content?

- Analyst 1: Module 1 (information comes up later in other parts)
Tools and Forms
Analyst 2: Depends on application (use what need)
Analyst 3: Group part (family, social)

Other general suggestions for use?

- Analyst 1: Don't use all at one time
Analyst 2: Let me see what I want to see and not get bogged down
Analyst 3: No

4. How could this course content be changed to more closely fit NASA needs?

- Analyst 1: Don't know much about how IBM does code inspection
If were to use, would need to follow IBM guidelines
Analyst 2: Are they teaching IBM practices
Provide "what if's"
Management suggestion too stiff, not realistic
Analyst 3:

5. What other content/processes do you think this type of instruction might be appropriate for? Be specific.

- Analyst 1: Communications classes
Learning a new language
Analyst 2: Anything
Training in labs can use this
Management
Development processes
Whole cycle

6. How realistic were the video scenarios during the instructional modules?

- Analyst 1: Yes, they were realistic
Analyst 2: Good representations
Analyst 3: Pretty realistic, but don't like attacks on producer

7. How realistic were the video scenarios during the simulation?

- Analyst 1: Yes, they were realistic
Analyst 2: Never been in one, but liked the idea
Analyst 3: Okay

System Capabilities

1. Did you have difficulties with any sections of the course?

- Analyst 1:** Pretty easy to use
Some problem with tools, but maybe didn't pay enough attention to module
Enough instruction in how to use tools
Natural language interface
- Analyst 2:** Natural language interface
Not consistent
Mouse
Should be easy and not distracting
- Analyst 3:** Occasional minor occurrence of what to do or click on next
No preference using either the mouse or keyboard

If yes, what areas of the course did you have difficulty with?

Analyst 1:

Analyst 2:

Analyst 3:

Do you think these difficulties could be overcome?

Analyst 1:

Analyst 2: Yes

Analyst 3:

What suggestions do you have for overcoming these difficulties?

Analyst 1: Help button (use video and audio to provide instruction) rather than text

Analyst 2:

Analyst 3:

2. What is your opinion of the ease of use in the following parts of the course?

Analyst 1:

Analyst 2:

Analyst 3:

Auditorium:

Analyst 1: Very good introduction of course

Analyst 2:

Analyst 3:

Training Room:

Analyst 1: Some modules got a little long

Analyst 2: Would like to be able to pause and back out

Needs better initial instruction in how to navigate

Define buttons and use consistently
Mouse interface (make hot spot bigger, position on location to click)
Analyst 3: Easy
Some waiting for audio, but not a major problem

Module 1 (Formal Inspections: Purpose and Process):

Analyst 1:

Analyst 2:

Analyst 3:

Module 2 (Inspections Types and Differences):

Analyst 1:

Analyst 2:

Analyst 3:

Module 3 (Inspection Roles and Pitfalls):

Analyst 1: Long

Analyst 2:

Analyst 3:

Module 4 (Inspections Tools and Forms):

Analyst 1:

Analyst 2:

Analyst 3:

Module 5 (Inspections Communications):

Analyst 1:

Analyst 2:

Analyst 3:

Library:

Analyst 1: Good to use
Good way to learn and get information
Analyst 2: Good, has lots of potential
Easy to use
Would like a sort function to find things
Analyst 3: Easy to use

Office:

Analyst 1:

Analyst 2:

Analyst 3: Easy to use
Liked tools
Tools were good

Conference Room (simulated code inspections):

Analyst 1: Not very much interaction (but not used to tools or code)
Provide chances to interact
Was short

Analyst 2:

Analyst 3: No real problem

APPENDIX B
RAW DATA
(OBSERVATIONS)

OBSERVATIONS
- Analyst 1 -

TASK	PROBLEMS/ DIFFICULTIES	ANALYST COMMENTS	OBSERVER COMMENTS	TIME SPENT (minutes)
Auditorium				8
Training Room: Module 1	<ul style="list-style-type: none"> • not clear on whether to click on arrow, repeat, or exit buttons • accessed help and could not get out 	<ul style="list-style-type: none"> • wanted to go back one section, but program did not allow at that point 	<ul style="list-style-type: none"> • asked if okay to take notes (did take some notes) • repeated section on measurement one time • reread summary screen before moving on • looked at process steps for a long time • repeated planning section • repeated section on overview • very attentive, quiet • repeated section on inspection defect list • a bit unclear about what to do at the end of the module • surprised to see Buzz • repeats many sections • reads screens carefully 	23

<p>Training Room: Module 2</p>			<ul style="list-style-type: none"> • repeated section on inspection functions • repeated section on walkthroughs (definition, functions) • rereads screens before continuing • repeated review/audit functions • repeated section on advantages and disadvantages of software inspections • repeated section on advantages and disadvantages of walkthroughs • repeated section on design review/audits 	<p>17</p>
---------------------------------------	--	--	--	-----------

Training Room: Module 3	<ul style="list-style-type: none"> program did not check off box on menu for having done all "what if's" (and at higher level for completing the moderator) problem in producer section with the software (VRAM possibly corrupted) problem with software, could not go on after an AV segment (had to click on pause and move on and then okay) (did not check section off at menu) 	<ul style="list-style-type: none"> I would prefer to take in smaller sections (tired) 	<ul style="list-style-type: none"> repeated section on reading code and recording defects repeated section on key responsibility of moderator went through all "what if's" for the moderator has stopped taking notes appears tired 	68
Training Room: Module 4				25
Training Room: Module 5			<ul style="list-style-type: none"> liked video segments in instructional modules liked scenarios in instructional modules very quiet and passive overall 	35
Posttest 1		<ul style="list-style-type: none"> requested a copy of the test to keep 		79
Library			<ul style="list-style-type: none"> very quick listened to introduction only, did not explore any further 	5

Office			<ul style="list-style-type: none">• very quick• did not practice with code or tools	5
Code Inspection Simulation	<ul style="list-style-type: none">• kept adjusting audio (inconsistency in level of audio)• difficulty in how to use the natural language interface• did not know how to continue• did not know how to look at code (trying to find name of a variable)• did not know how to page up/page down	<ul style="list-style-type: none">• not familiar with ADA• not clear on how to use the natural language interface	<ul style="list-style-type: none">• user prompted interaction one time• program prompted interaction three times• did not interact for first time until the program prompted to do so• not interacting• missed noting an error• real trouble using the natural language interface• not familiar with ADA• took a long time filling out the first recorder form• very attentive• did not stop the flow when others got off the topic (basketball game)• did not appear that the program had prepared the analyst very well to use the natural language interface• did not classify any errors• commented that there was not much interaction	30

Analyst Questionnaire	• did not understand Likert scale			15
Interview Questions				45
Posttest 2				70

OBSERVATIONS
- Analyst 2 -

TASK	PROBLEMS/ DIFFICULTIES	ANALYST COMMENTS	OBSERVER COMMENTS	TIME SPENT (minutes)
Auditorium		<ul style="list-style-type: none"> • too fragile (clicked on exit accidentally and exited) • video was entertaining • liked transition between airline ticket operator and president of company • liked it • good video 		10

<p>Training Room: Module 1</p>	<ul style="list-style-type: none"> • unclear which buttons to use when • unclear how to go on (exit button) 	<ul style="list-style-type: none"> • did not know what to do when got into the floor-plan (program needs to direct you what to do) • would like to be able to pause at any time (if interrupted or to take notes) • more navigation information (how to proceed with arrow key) • button should say "continue" • instruction clear and to the point so far • tells four roles and uses in audio, but haven't been defined yet • have mouse come up on arrow (so don't have to move it every time) • puts up code inspection summary form, but does not describe and has a slightly different title) • unclear what "keep accurate statistics" means • unclear what "stick to the standard or change it" means • backwards arrow looks like it should mean repeat rather than go back one menu 	<ul style="list-style-type: none"> • repeated one section (to see if button worked) • tested the "Stop AV" button • stayed on screen to read information after audio was finished • used "Stop AV" • repeated section 	<p>25</p>
--------------------------------	---	--	--	-----------

<p>Training Room: Module 2</p>			<ul style="list-style-type: none"> • buttons always in same location • larger hot spot • tell how to use buttons at beginning of program • "Buzz" was funny 	
			<ul style="list-style-type: none"> • words in audio don't match text very well • whole card (button) in card file should be active, not just title • define who the "external people" are better • doesn't talk about disadvantages (it is on screen, but not in audio) • likes having advantages and disadvantages listed out • no audio with first screen (not consistent with other screens) • help is lacking • mouse is very distracting (not in correct position, have to move each time) • last section in card file is lacking in content (only one card which is not consistent with other cards) 	15

<p>Training Room: Module 3</p>	<ul style="list-style-type: none"> • no audio with first screen (puzzled as to what to do, inconsistent with other parts of program) • missed one screen when hit "Pause AV" • "Critical Error" (after paused for five minutes) (clicked on "retry" and worked fine) • clicked mouse two times accidentally and got corrupted video (there was a long pause in the program and analyst wondered if had clicked the key or not, so did it again) 	<ul style="list-style-type: none"> • content is unclear (moderator role in planning) • confusing because the audio is talking about the producer in the section on the moderator • talks about reader and recorder during the moderator section • information about keeping people on an even keel and working together is good information • like scenarios (hopefully there are more of these) • learn more from examples than from just text • doesn't like "Pause AV" (might miss something when it continues on) • this program is much better than sitting in front of a manual • some screens continue without you being able to decide when to move (not consistent) • in the "what if" section some of the consequences are too severe (send to management) (may not be realistic and should not convey this to new hires) 	<ul style="list-style-type: none"> • liked it when he saw a video segment for this first time ("Oh, this should be good.") • laughing • appears very attentive during the scenarios • laughs and nods head throughout scenarios as if he can relate to what is happening • only went through four "what if's" (not all four pages) • appeared a little confused when audio took a long time to load 	<p>80</p>
--------------------------------	---	---	---	-----------

			<ul style="list-style-type: none"> • what are "entry" and "exit" criteria • checklists before, during, and after are nice (would like a hardcopy to use during an inspection) (good reference pages) • interesting the producer picks the moderator (couldn't this lead to problems?) • something in program here clarified what the analyst wanted an answer to earlier in the program • different use of buttons (back arrow and forward arrow on same screen and no repeat button) • no audio with text screen (inconsistent) • same piece of video as seen before • video (surprised at female scolding members) • talking about recorder in reader section (confusing) • even if experienced, this section on roles may be a helpful reminder 		
--	--	--	--	--	--

		<ul style="list-style-type: none">• no data filled in on the form (as audio says there is)• moderator tone of voice (implies a hierarchy, not a peer relationship)• another course the analyst took was much more interactive (used scenarios)• this is much better than reading a manual• would like to go to an inspection (see how members can fulfill roles and still participate on a technical level)		
--	--	---	--	--

<p>Training Room: Module 4</p>	<ul style="list-style-type: none"> • program continues to next screen without enough time to complete reading the current screen (inconsistent with rest of program) 	<ul style="list-style-type: none"> • need more guidance from the program about what this part is and how to use it • tired of mouse (real hassle) • did not know what the code in the window was or what to do with it (show me what you are talking about such as how to pull down windows) • program does not tell you enough to get a good feel for what tools are and when they are available to you • program goes through the "show" part of the pull down menus too fast • the "trace" function is unclear (does it stop on the line or before, this could change the value of the variable) • question about the availability of tools (misleading whether only for simulation or available in reality) 		<p>25</p>
--------------------------------	---	--	--	-----------

		<ul style="list-style-type: none"> • would be nice to use own code and own tools • likes content of videos • if something is only for the sake of the simulation, why make it so complex 		
--	--	---	--	--

<p>Training Room: Module 5</p>	<ul style="list-style-type: none"> • after section on groups, no back arrow to take you a level up to go through the next section on natural language interface (inconsistent with the rest of the program) 	<ul style="list-style-type: none"> • too long before audio starts when there is a picture and text on the screen • "pause" button would be better then "Stop/Move On" • most content on groups was obvious, did not really learn anything • information on groups was out of place • some information on groups was useful, but didn't really apply to code inspection • feels like I'm taking a Sociology course • important to differentiate between groups, but do it much quicker • some screens are just text and read word for word • use video sequences for group instruction • get to the point (technical people and engineers will get bored) • need to deal with this topic, but there needs to be a better way to do it • too clinical (make more fun or use more examples) 	<p>40</p>
--------------------------------	--	--	-----------

			<ul style="list-style-type: none"> • more emphasis on "listening skills" • examples should include a correct and incorrect one • "conflict tampered with respect" should be "tempered" • likes caricatures (entertaining) • moderator has lots to do • likes when says go to library for more information • unclear on the purpose of the natural language interface • what is the difference between "passive" and "neutral" in the natural language interface • will be interesting to see how easy this interface is to use • the natural language interface is a neat idea if it works 		
--	--	--	--	--	--

Posttest 1			<ul style="list-style-type: none"> • previous experience helped answer some questions (rather than the instruction) • multiple choice last time helped answer some questions this time • some content faded (overload) 		60
Library			<ul style="list-style-type: none"> • library is nice (like the idea) • nice to have access to training material • Buzz is fun and good distractor • really like video segments • easy to highlight relevant articles (like this feature) • lot better than manuals 		5
Office			<ul style="list-style-type: none"> • hard to use tools (too much information earlier, overloaded and did not get enough information about it) • help needs to be more sensitive to how to use the screens and tools • need better instructions on how to use this part of the program 		5

Code Inspection Simulation	<ul style="list-style-type: none"> • didn't really understand how to use the natural language interface • frustration with creating sentences using the natural language interface 	<ul style="list-style-type: none"> • not enough instruction on how to use the natural language interface • instructions earlier were overwhelming • pretty neat to be able to do this • I need to use other comments (rather than "let's move on") • this is good • when choosing from the natural language interface menu, have block highlighted I should choose from (subject and then predicate) • needs to be more user friendly • really good, I really like this idea 	<ul style="list-style-type: none"> • user prompted interaction twelve times • program never prompted interaction 	35
Analyst Questionnaire				10
Interview Questions				25
Posttest 2				25 * did not take Part 2

OBSERVATIONS
- Analyst 3 -

TASK	PROBLEMS/ DIFFICULTIES	ANALYST COMMENTS	OBSERVER COMMENTS	TIME SPENT (minutes)
Auditorium				10
Training Room: Module 1	<ul style="list-style-type: none"> unclear which button to use at end of section where does the "Exit" button go 	<ul style="list-style-type: none"> back arrow looks like a "repeat" arrow rather than one taking you back one menu level "I guess its the exit button, that's the only choice." 	<ul style="list-style-type: none"> no problem finding arrow to continue on first page (able to navigate without any help) moving through instruction quickly does not take additional time to read or review screens no comments, moves through quickly 	15
Training Room: Module 2		<ul style="list-style-type: none"> tried to click on quality assurance techniques because no audio with screen (inconsistent) doesn't know what terms mean 	<ul style="list-style-type: none"> no comments, moves through quickly 	10

<p>Training Room: Module 3</p>	<ul style="list-style-type: none"> extra "the" in the text (p. 2 of 4, #4) 	<ul style="list-style-type: none"> no audio with first screen was confusing audio and text on screen don't match well enough (confusing and hard to follow) talking about the reader and recorder in the moderator section (seems out of place) don't need to read checklists now, assume they are a summary prefer audio with text (retain more) don't really feel I need more information on buttons and how to navigate same video as before (used "Stop AV" to move on) same video again (used "Stop AV" to move on) like motion video scenarios like "Stop AV/Move On" feature mouse moves around on screen without me moving it (this is bothering some) 	<ul style="list-style-type: none"> appears a little confused if audio is not played immediately moving through quickly surprised when saw a screen full of text did not read three checklists in entirety appeared to be waiting for audio on screen without audio did not sit through two videos already seen earlier 	<p>55</p>
--------------------------------	---	---	---	-----------

Training Room: Module 4		<ul style="list-style-type: none"> • section was too quick (couldn't read screen before moving on) • unclear when or where to use tools 		15
Training Room: Module 5		<ul style="list-style-type: none"> • unclear where this section on groups is going • what does this have to do with code inspections • include lesson family and social groups and more on what is directly related to code inspection • information on groups should be in a different course (not related to code inspection) • this is the hardest part to stay interested in • unclear what the instruction on the natural language interface is for (purpose) or when it will be used 	<ul style="list-style-type: none"> • appeared very bored with group section (didn't see relevance of the section) • somewhat passive throughout the instructional modules 	30
Posttest 1				55
Library		<ul style="list-style-type: none"> • looks good • has potential 		5

Office		<ul style="list-style-type: none">• good tools except more documentation is needed• confused as to the purpose of this section			10
Code Inspection Simulation	<ul style="list-style-type: none">• some trouble using the natural language interface, but was able to construct logical sentences	<ul style="list-style-type: none">• pretty easy to use the simulation• seems slow for the other participants to react• confused as to why the producer reacted in a certain way• fun overall• for a new person who has never attended a code inspection, this would be very helpful	<ul style="list-style-type: none">• user prompted interaction twelve times• program never prompted interaction• did a good job of creating more complex sentences• participated fully, very interactive		35
Analyst Questionnaire					7
Interview Questions					10
Posttest 2					39

APPENDIX B
RAW DATA
(DEMOGRAPHIC INFORMATION)

DEMOGRAPHIC INFORMATION

Title: Analyst 1: Associate Programmer
Analyst 2: SSW Engineer
Analyst 3: Software Engineer

Degree(s): Analyst 1: Computer Science
Analyst 2: BS Mechanical Engineering
Analyst 3: BS Electrical Engineering

Computer Experience

1. How many years have you used computers?
Analyst 1: 5 years
Analyst 2: 12 years (PC's for basic technical computations and word processing)
Analyst 3: 14 years
2. How many hours per day do you currently use computers in your work?
Analyst 1: 6 hours
Analyst 2: 1-2 hours per day
Analyst 3: 4 hours
3. For what applications do you currently use computers (programming, word processing, etc.)?
Analyst 1:
Analyst 2: Word processing/documentation, PROFS, storyboard graphics presentations
Analyst 3: Word processing, EMail, graphics, information system

Programming Experience

1. How many college level courses have you taken where you were required to write or understand a program in a procedural programming language such as Pascal, FORTRAN, C, Ada, etc?
Analyst 1: 7 classes
Analyst 2: 5-6 (Fortran and Pascal)
Analyst 3: Fortran
2. How many college-level courses have you taken where you were required to write or understand a program in the Ada programming language? Briefly describe your level of experience with the Ada programming language.
Analyst 1: None
Analyst 2: None
Analyst 3: No college level course, one week course (40 hours) on Ada

3. How many computer languages can you understand and program with? Please list these languages.
- Analyst 1: Pascal, C, Cobol, LSP
Analyst 2: Basic, Fortran, Assembler (basic programs)
Analyst 3: Fortran, Basic, C, Ada (weak in C & Ada)

Code Inspection Process

1. Have you ever participated in a code inspection before? If yes, how many?
- Analyst 1: 4
Analyst 2: No
Analyst 3: No
2. How long have you been involved in the code inspection process?
- Analyst 1: 2 months
Analyst 2:
Analyst 3: 0
3. Is code inspection part of your present job?
- Analyst 1: Yes
Analyst 2: Only when I need to look at code to troubleshoot a lab problem
Analyst 3: No
4. When was the last time you participated in a code inspection? What roles did you perform?
- Analyst 1: Spring 1992 (optional attendee: code review)
Analyst 2: Never
Analyst 3: Never
5. Have you ever received formal training in the code inspection process? If yes, please describe briefly.
- Analyst 1: No
Analyst 2: No
Analyst 3: No
6. Have you taken any college level courses where software engineering concepts were taught? If yes, how many? Please describe briefly.
- Analyst 1: 1 class. There is one project for the whole class. The project is broken up into small parts so that each group is responsible for it.
Analyst 2: No
Analyst 3: No, only courses offered at work

7. Have you taken any college level courses where software technical reviews were discussed? If yes, how many? Please describe briefly.
- Analyst 1: Software engineering, made presentations on design and code implementation
Analyst 2: No
Analyst 3: No

General Information

1. Have you ever received training via the computer before (computer-based training)?
- Analyst 1: Yes
Analyst 2: Yes
Analyst 3: Yes

If yes, how many courses and for what topics?

- Analyst 1: 1 course (Flight Control, GNC)
Analyst 2: Situational Leadership, Intro to Assembler
Analyst 3: 8 courses (GPC Synchronization, Bus Reconfiguration, PASS ILoad Recon, Ascent Overview, GN&C, Crew SW Interface, CRT Display Overview, PASS Architecture)

Do you feel like it was an effective way to learn?

- Analyst 1: Yes
Analyst 2: Yes
Analyst 3: It was OK

Did you like learning from computer-based training?

- Analyst 1: Yes
Analyst 2: Yes
Analyst 3: It was OK

What do you feel are some strengths of computer-based training?

- Analyst 1: Interactive, easy to review, look up terminology easy, know where you are and test helps to reinforce the ideas
Analyst 2: Graphics capability, flexible to personal schedule, multimedia tools can be used
Interactive sessions are great
Analyst 3: Not as boring as reading from a manual
More effective examples can be provided

What do you feel are some weaknesses of computer-based training?

- Analyst 1: Slow
Analyst 2: They are only as good as the programmer makes it. The programmer needs to clearly answer the key issues and questions. It can be limiting.
Analyst 3: Nobody to answer you questions

2. Have you ever received any training which incorporates audio and video images? If yes, please describe.

Analyst 1: No

Analyst 2: Yes. The Situational Leadership class was interactive and very effective. I have also seen some multimedia Shuttle presentations and am trying to develop some training stories on Storyboard.

Analyst 3: No

3. Have you ever received any training which incorporates a process simulation, for example code inspection, before? If yes, what process was taught? Please describe briefly.

Analyst 1: No

Analyst 2: No

Analyst 3: No

APPENDIX C
ANNOTATED COURSE OBJECTIVES

OBJECTIVES FOR THE CODE INSPECTION COURSE
Advanced Learning Technologies Project
Software Engineering Institute
Carnegie Mellon University

All items in bold were provided as objectives by Carnegie Mellon University; other information was added by SwRI after reviewing the program.

TRAINING ROOM OBJECTIVES

Overall Training Objective:

To provide skills and knowledge that software engineers will need to conduct successful software inspections within the interactive system that will transfer into real-life inspection environments, to efficiently use the resources and tools within the DVI system, and to experience ease in using the system for learning enjoyment.

Module 1 Objectives - Formal Inspections: Purpose and Process

The Software Engineer will be able to:

- 1. Describe the purpose of formal inspections.**
 - A. Set a standard of excellence**
 - B. Promote correctness and completeness**
 - C. Promote adherence to project style and rules of construction**
 - D. Promote compliance with technology practices**
 - E. Provide structured ways to view product systematically**
 - F. Obtain metrics for project management and process control**
- 2. Identify the stages of the formal inspection process.**
 - A. Planning (assigning tasks)**
 - B. Overview (communications/education)**
 - C. Preparation (education)**
 - D. Inspection (find errors)**
 - E. Reporting (report errors)**
 - F. Rework (fix errors)**
 - G. Follow-Up (ensure correct fixes)**
 - H. Reinspect (find final errors)**
- 3. Describe the benefits of the formal inspection process.**
 - A. Cost savings**
 - B. Improve error detection**
 - C. Reduce cost to customer**
 - D. Improve productivity**
 - E. Increase product knowledge**
 - F. Improve process control**

4. Describe the role of planning and preparation in the inspection process.
 - A. Planning
 - workproduct meets entry criteria
 - moderator selected
 - decision on overview
 - inspectors selected and assigned roles
 - overview and inspection meetings scheduled
 - B. Overview Session
 - often led by producer
 - educational for team
 - background information given on work product
 - C. Preparation
 - workproduct must be thoroughly reviewed prior to inspection meeting
 - individual preparation to locate possible defects and gain knowledge of workproduct
5. List the review roles assumed during the inspection.
 - A. Moderator
 - B. Reader
 - C. Producer
 - D. Recorder
6. List the types of checklists that can be used before, during the inspection and for followup.
 - A. Rules of Construction
 - B. Correctness Checklist
 - C. Style Checklist
 - D. Metrics Checklist
 - E. Technology Checklist

*** Additional Information**

Definition of an inspection:

- a small peer group process whose purpose is the detection and correction of software product defects
- rigorous entry and exit criteria
- a formal procedure for identification, report and rework of workproduct defects

Characteristics of an inspection meeting:

- no more than 2 hours
- initiated by moderator
- preparation times recorded
- reader guides the group
- producer helps identify defects
- recorder records defects on Inspection Defect Log

Basic rules for Code Inspections:

- management should not be present at inspections
- inspections are not a tool for worker evaluation
- producers should not be moderator, reader, or recorder on their own work
- checklists of questions can be used to define the task and stimulate defect finding
- inspections should be limited to 2 hours
- producers should not spend more than 25% of their time in inspection-related duties

Guidelines for Successful Code Inspections:

- be prepared
- be willing to associate and communicate
- have at least one positive comment
- find defects, not solutions
- stick to the standard or change it
- do not use derogatory language
- record all issues in public
- evaluate the product, not the producer
- stick to the technical issues
- keep accurate statistics

Module 2 Objectives - Inspection Types and Differences

The Software Engineer will be able to:

- 1. Discriminate between inspections, walkthroughs and design reviews/audits.**
 - **Software Inspection:** small group process whose purpose is the detection & correction of software product defects.
 - rigorous entry & exit criteria
 - process management tool for improving quality
 - **Internal Walkthroughs:** a dynamic presentation of a software product usually presented by the developer of the software to a peer group for the purpose of improving the quality of the work product.
 - vary in format from very informal to structured reviews
 - **Formal Design Reviews/Audits:** an agent external to the process being examined.
 - insures proper validation
 - insures that producing intended results
- 2. Describe the different functions of formal inspections, walkthroughs and design reviews/audits.**
 - **Software Inspection:**
 - small trained peer group
 - specific formal agenda
 - specific roles
 - function to identify, classify, & report defects
 - process control tool
 - rigorous entry/exit product criteria
 - product examined at defined checkpoints
 - **Internal Walkthroughs:**
 - large or small peer groups
 - informal to structured
 - early defect detection
 - educational support
 - **Formal Design Reviews/Audits:**
 - external process review
 - customer, user, & developer usually involved
 - affirms or negates status of product
 - not used for defect detection
- 3. Compare the differences between inspection and walkthrough procedures.**
 - **Software Inspection**
 - **Advantages**
 - formality yields consistent results
 - early quantitative quality evaluation
 - historical error database to reduce recurrences
 - **Disadvantages**
 - keyed to developer's viewpoint
 - **Internal Walkthroughs**
 - **Advantages**
 - collective review and detection of defects
 - early detection of defects

- Disadvantages
 - varying rigor yields inconsistent results
 - little "corporate memory" to reduce recurrences
- Formal Design Reviews/Audits
 - Advantages
 - integrate developer, user, and customer views
 - furnish management visibility for approval or disapproval of proceeding to next phase
 - Disadvantages
 - seldom challenge technical basis of design
 - not effective for quality evaluations

4. List other techniques for assuring software quality.

- A. Automatic Tools Checking
- B. Team Leader Checking
- C. Simulation
- D. Prototyping
- E. Unit, Subsystem, and System Testing

Module 3 Objectives - Inspection Roles and Pitfalls

The Software Engineer will be able to:

- 1. Describe the roles of moderator, reader, recorder and producer in formal inspections.**
 - A. Moderator**
 - responsible for verifying entry and exit criteria
 - making sure everyone is prepared and contributing
 - making sure reviewers do not go off on tangents during the inspection
 - making sure that the focus of the inspection remains on the code and not the producer
 - B. Reader**
 - responsible for letting everyone know what is being discussed
 - pacing the meeting
 - introducing and summarizing the next piece of code to be discussed
 - C. Recorder**
 - responsible for writing down defects found during the inspection
 - classifying errors according to predefined categories
 - noting action items to take care of after the review is complete
 - D. Producer**
 - writer of the code being inspected
 - answer any specific questions about the code
 - present information about the code without getting defensive when the code is questioned for defects
- 2. Describe special problems often faced by moderators in conducting a software inspection.**
 - A. Attack on producer**
 - B. Moderator dominates**
 - C. Followup communications**
 - D. Pitfalls**
- 3. Describe in detail the roles of the manager, moderator, reader, recorder and producer at each step in the formalized inspection process.**
 - A. Planning**
 - 1. Manager**
 - involved
 - 2. Moderator**
 - selected from unrelated project by producer or first line manager
 - verifies with producer that workproduct meets entry criteria
 - 3. Reader**
 - 4. Recorder**
 - 5. Producer**
 - entry criteria
 - clean compile/assembly with time tags
 - functions descriptions and comments
 - detailed design materials
 - change request (if appropriate)
 - support documentation

B. Overview

- 1. Manager**
- 2. Moderator**
 - schedules meetings
 - makes physical arrangements
 - sends notice of meeting time and place
 - makes sure all members get materials needed for preparation
 - gets confirmation of members' acceptance of schedule and receipt of materials
- 3. Reader**
 - must be familiar with workproduct so as to be able to paraphrase the workproduct in detail
- 4. Recorder**
 - education regarding workproduct
 - learn classification of defects
- 5. Producer**
 - assemble pertinent documentation
 - presents and educates group regarding workproduct
 - producer provides tutorial on specialized design or implementation technique

C. Preparation

- 1. Manager**
- 2. Moderator**
 - study workproduct
 - note defects
- 3. Reader**
 - study the workproduct and specifications documents
 - organize a strategy for paraphrasing the workproduct
- 4. Recorder**
 - study the workproduct
 - identify defects
- 5. Producer**
 - review workproduct prior to inspection meeting

D. Inspection Meeting

- 1. Manager**
- 2. Moderator**
 - introduces team
 - states purpose of the meeting
 - checks for changes in baseline
 - checks all materials provided
 - records preparation times
 - determines preparedness to continue
 - keeps group on target and meeting objectives
 - determines disposition of workproduct
- 3. Reader**
 - paces the group and guides the group by paraphrasing the code
 - keeps track of location of issues and refocuses group on relevant parts of product
 - reader knows workproduct and paraphrases segments
- 4. Recorder**
 - notes location, description, class, and type of defect
 - recorder needs technical awareness of workproduct
 - needs good judgment and ability to classify defects
 - all issues must be recorded completely and accurately

5. Producer
 - participates as a reviewer and raises issues about the workproduct
 - acts as an inspector and identifies defects
 - adopts non-defensive attitude
- E. Reporting
 1. Manager
 2. Moderator
 - closes inspection meeting (if pass)
 3. Reader
 4. Recorder
 - all issues must be recorded completely and accurately
 - fills out inspection defect list
 5. Producer
- F. Rework
 1. Manager
 - involved
 2. Moderator
 - verifies defect corrections made
 3. Reader
 4. Recorder
 5. Producer
 - performs rework
 - corrects defects listed on the Inspections Defect List
 - producer along with moderator helps resolve open issues
- G. Followup
 1. Manager
 2. Moderator
 - completes inspection management report
 3. Reader
 4. Recorder
 5. Producer
 - producer consults with moderator to verify that corrections have been completed
 - moderator handles reporting to management that corrections are complete
- H. Reinspection
 1. Manager
 2. Moderator
 - schedules reinspection (same as rescheduling the initial inspection)
 - completes physical arrangement
 - sends notice time and place
 - provides materials
 - inspectors confirm schedules
 3. Reader
 - attends when scheduled
 4. Recorder
 - attends when scheduled
 5. Producer
 - participates as an inspector of the workproduct
 - producer helps locate final errors

4. Identify the checklists and forms used by each participant in the formal inspection process.

Forms:

- A. Preparation Log
 - completed by all inspectors and given to the moderator as a record of the preparation done for the inspection
- B. Inspection Defect List
 - completed by the recorder during the inspection as a record of points brought up during review
- C. Code Inspection Summary Report
 - completed by the recorder following the inspection from data collected in the inspection defect list; passed on to the moderator when finished
- D. Management Summary Report
 - completed by the moderator following the inspection, incorporating information from the Code Inspection Summary Report; given to management when finished

5. Discriminate among potential problem situations emerging from interaction within the group during inspection.

Helpful Hints (moderator) (What if):

- a quiet person hasn't spoken yet?
- someone talks too much?
- someone is too aggressive?
- everyone isn't prepared?
- someone tries to rush through the inspection?
- someone has really been obstructive during the inspection?
- the meeting drifts into irrelevant subjects or unnecessary detail?
- you haven't assembled the materials needed for the inspection team?
- the product being inspected isn't very good?
- a good inspection wasn't obtained?
- etc. (4 screens)

6. Identify with the "model behavior" of each participant in the inspection.

Key Responsibilities:

- A. Manager
 - Planning
 - Rework
- B. Moderator
 - Overview
 - Preparation
 - Code Inspection Meeting
 - Reporting
 - Follow-up
 - Reinspection
- C. Reader
 - Overview
 - Preparation
 - Code Inspection Meeting
 - Reinspection
- D. Recorder
 - Overview
 - Preparation
 - Code Inspection Meeting

- Reporting
- Reinspection
- E. Producer
 - Overview
 - Preparation
 - Code Inspection Meeting
 - Rework
 - Follow-up
 - Reinspection

*** Additional Information**

A Good Inspection:

- accurate assessment of the workproduct
- defects detected efficiently

Disposition Categories:

- Pass (meets exit criteria)
- Does Not Pass (rework, reinspect)

Defect:

- non-compliance with a product specification or document standard
- defect classes (Fagan)
 - M - Missing (material called for in specs, but not included)
 - E - Extra (exceeds specifications)
 - W - Wrong (material is present, but contains flaw)
- generic set of defect classes
 - DE - design error
 - IN - interface
 - DA - data
 - LO - logic
 - PF - performance
 - IO - input/output
 - CC - code comment
 - ST - standards
 - DC - documentation
 - SN - syntax

Module 4 Objectives - Inspection Tools and Forms

The Software Engineer will be able to:

- 1. Demonstrate the use of the computer tools (debugger, code analysis, and notetaking) for analyzing a piece of Ada code.**

Code Inspection Tools:

- hypertext tools
 - help consistency checking between documents
 - keep track of notes
- code debugger tools
 - help check correctness of ADA code
 - enable better understanding of the code to be inspected

- 2. Prepare for the simulated code inspection by analyzing and taking notes on an Ada code sample.**

- 3. Describe how checklists and report forms are used before, during, and in follow-up to the formal inspection process.**

Checklists:

- rules of construction
- correctness checklist
- style checklist
- metrics checklist
- technology checklist

Report Forms:

- preparation log
- inspection defect list
- code inspection summary report
- management summary report

- 4. Demonstrate use of the recording form and summary form.**
- 5. Describe the process for notifying team members about the review when assuming the role of moderator.**
 - schedules meetings
 - makes physical arrangements
 - sends notice of meeting time and place
 - makes sure all members get materials needed for preparation
 - gets confirmation of members' acceptance of schedule and receipt of materials
- 6. Describe the process for accessing the computer tools within the "office" environment at Ultimex.**
 - use tools in office and training room
 - pop-up menu (left mouse button)
 - move/resize window (right mouse button)

*** Additional Objectives (from Carnegie-Mellon)**

- 7. Appreciate the value of software inspections as an effective technique for improving software quality.**

- 8. Acknowledge the importance of inspecting software as an organizational approach to cost reduction and improved productivity.**
- 9. Value the impact of controlling for software techniques.**

Module 5 Objectives - Inspection Communications

The Software Engineer will be able to:

- 1. Demonstrate the use of the interface tool for communicating with the simulated inspections group.**

Task Interface Summary (Conversational Interface)

Purpose: to create a sentence to say to the other reviewers during a code inspection

Consists of:

- talk menus
 - code window
 - specifications window
 - notes window
 - sentence window
 - emotions icons
 - reviewers icon
- To enter the talk interface, click a mouse button while another reviewer is talking. You will given a prompt that you will be given the floor shortly.
 - To exit the talk interface, click the mouse on the image of the reviewers.
 - The code being inspected, specifications for this code, and your notes about the code are accessible via text windows in the talk interface.
 - Remember that if you need assistance while in the talk interface, you can always access the help icon.

- 2. Show how the attitudinal attributes (icons and phrases) are selected and used for effective emotional context in communication with the simulated group.**

Icons: blue - yellow (neutral) - red

Phrases: carry attributes

- 3. Recognize the existence of group experience within his/her own life pattern.**
- 4. Accept the importance of group dynamics as a human interactive communications skill.**
- 5. Define what a group is and why we function as groups.**
 - its membership can be defined
 - it possesses a group consciousness
 - it possesses a shared sense of purpose
 - its members have an interdependence in the satisfaction of their needs
 - interaction among the members is evident and the group is able to act in a uniting manner
 - interaction and communications are necessary in order to reach the shared goal, and decisions must be agreed to by at least a majority of the members of the group
- 6. Identify common problems within a task-oriented group.**
 - "detrimental conflict"
 - types of members
 - aggressive
 - silent
 - abusive
 - rambling
 - snapping (witty)

7. Discriminate between a social group and a task-oriented group.

Types of Groups:

- social
 - organized in an informal way
 - main purpose is social in nature
- family
 - nucleus for learning, love, trust, intimacy, acceptance and self-worth
 - forms the basis for behaviors
- task-oriented
 - achieves goals through effective communication
 - participants more likely to accept decision results because they were actively involved in the group problem solving
 - comes together to accomplish goal or task
 - heart of any effective organization

8. Identify the characteristics of a successful task-oriented group meeting and how these same characteristics apply to the formalized software inspection.

- company policy
- meeting has structure and agenda
- preparation done prior to meeting
- effective interpersonal communication
- no interruptions
- focus for conclusions and follow-up

9. Recognize the non-verbal and verbal messages which signal problems within a group meeting.

10. Use group process skills to effectively communicate with the simulated members of the inspections group.

*** Additional Objectives (from Carnegie-Mellon)**

11. Examine the positive and negative aspects of his/her participation in the simulated inspection, and determine the aspects of his/her role which influenced the inspection outcome, through review of the feedback presented during the inspection simulation as well as from the follow-up progress report.

Manager Track Objectives

The Manager will be able to:

- 1. Identify the role of the manager in the inspection process.**
- 2. Describe the purpose of formal inspections.**
- 3. Identify the stages of the formal inspection process.**
 - Module 1, Question 2
- 4. Cite the advantages of inspection versus walkthroughs.**
 - Module 2, Question 3
- 5. Describe the way in which inspection data can be used in future software development planning.**
- 6. Identify the key features of formal inspections that contribute to cost savings and error reduction.**
 - Module 1, Question 3, etc.
- 7. Cite the studies that support the use of formal inspections within organizations.**
 - see library articles
- 8. Describe a process for implementing formal inspections within an organization.**
- 9. Describe the key philosophical aspects of implementing inspections within an organization.**

System Use Objectives (from Carnegie Mellon)

- 1. Independently access all resources within the code inspection course, including the library, training room, and context-sensitive help system.**
- 2. Effectively analyze the code and type in defects ideas into the notes window, for later retrieval during the inspection simulation.**

APPENDIX D
CARNEGIE MELLON COURSE OBJECTIVES

Objectives for the Code Inspection Course

**Advanced Learning Technologies Project
Software Engineering Institute
Carnegie Mellon University**

Inspection Learning Objectives:

The software engineer will be able to:

- 1. Describe the purpose of formal inspections.**
- 2. Identify the stages of the formal inspection process.**
- 3. Describe the benefits of the formal inspection process.**
- 4. Describe the role of planning and preparation in the inspection process.**
- 5. List the review roles assumed during the inspection.**
- 6. List types of checklists that can be used before, during, and after the inspection.**
- 7. Discriminate between inspections, walkthroughs, and design reviews/audits.**
- 8. Describe the different functions of formal inspections, walkthroughs, and design reviews/audits.**
- 9. Compare the process differences between inspections and walkthroughs.**
- 10. Describe the roles of moderator, reader, recorder, and producer in formal inspections.**
- 11. Describe the special problems often faced by moderators in conducting a software inspection.**
- 12. Describe in detail the role of the manager, moderator, reader, recorder, and producer at each step in the formalized inspection process.**
- 13. Identify the checklists and forms used by each participant in the formal inspection process.**
- 14. Appreciate the value of software inspections as an effective technique for improving software quality.**
- 15. Acknowledge the importance of inspecting software as an organizational approach to cost reduction and improved productivity.**
- 16. Value the impact of controlling for software defects.**
- 17. Understand the importance of group dynamics as a human interactive communication skill.**
- 18. Define what a group is and why we function as groups.**
- 19. Identify common problems within a task-oriented group.**
- 20. Discriminate between a social group and a task-oriented group.**

21. Identify the characteristics of a successful task-oriented group meeting and how these same characteristics apply to the formalized software inspection.
22. Recognize the nonverbal and verbal messages which signal problems within a group meeting.
23. Examine the positive and negative aspects of his/her participation in the simulated inspection, and determine the aspects of his/her role which influenced the inspection outcome, through review of the feedback presented during the inspection simulation as well as from the follow-up progress report.

System Use Objectives:

The software engineer will be able to:

1. Independently access all resources within the code inspection course, including the library, training room, and context-sensitive help system.
2. Demonstrate skill using the source level debugging and hypertext tools in the office environment for analyzing code samples prior to the inspection of that code.
3. Effectively analyze the code and type in defects ideas into the notes window, for later retrieval during the inspection simulation.
4. Demonstrate the use of the interface tool for communicating with the simulated inspections group, including the effective use of the attitudinal attributes, which provide emotional context.

TRAINING ROOM OBJECTIVES

Over All Training Objective

To provide skills and knowledge that software engineers will need to conduct successful software inspections within the interactive system that will transfer into real life inspections environments, to efficiently use the resources and tools within the DVI system and to experience ease in using the system for learning enjoyment.

Module 1 Objectives — Formal Inspections: Purpose and Process

The Software Engineer will be able to:

1. Describe the purpose of formal inspections
2. Identify the stages of the formal inspection process
3. Describe the benefits of the formal inspection process
4. Describe the role of planning and preparation in the inspection process
5. List the review roles assumed during the inspection
6. List types of checklists that can be used before, during the inspection and for follow-up

Module 2 Objectives — Inspections Types and Differences

The Software Engineer will be able to:

1. Discriminate between inspections, walkthroughs and design reviews/audits
2. Describe the different functions of formal inspections, walkthroughs and design reviews/audits
3. Compare the differences between inspection and walkthrough procedures
4. List other techniques for assuring software quality

Module 3 Objectives — Inspection Roles and Pitfalls

The Software Engineer will be able to :

1. Describe the roles of moderator, reader, recorder and producer in formal inspections
2. Describe special problems often faced by moderators in conducting a software inspection
3. Describe in detail the role of the manager, moderator, reader, recorder and producer at each step in the formalized inspection process
4. Identify the checklists and forms used by each participant in the formal inspection process
5. Discriminate among potential problem situations emerging from interaction within the group during inspection
6. Identify with the "model behavior" of each participant in the inspection

Module 4 Objectives — Inspections Tools and Forms

The Software Engineer will be able to:

1. Demonstrate the use of the computer tools (debugger, code analysis, and notetaking) for analyzing a piece of Ada code
2. Prepare for the simulated code inspection by analyzing and taking notes on an Ada code sample
3. Describe how check lists and report forms are used before, during, and in follow-up to the formal inspections process
4. Demonstrate use of the recording form and summary form
5. Describe the process for notifying team members about the review when assuming the role of moderator
6. Describe the process for accessing the computer tools within the "office" environment at Ultimex

Module 5 Objectives — Inspections Communications

The Software Engineer will be able to:

1. Demonstrate the use of the interface tool for communicating with the simulated inspections group
2. Show how the attitudinal attributes (icons and phrases) are selected and used for effective emotional context in communication with the simulated group.
3. Recognize the existence of group experience within his/her own life pattern
4. Accept the importance of group dynamics as a human interactive communications skill
5. Define what a group is and why we function as groups
6. Identify common problems within a task-oriented group
7. Discriminate between a social group and a task-oriented group
8. Identify the characteristics of a successful task-oriented group meeting and how these same characteristics apply to the formalized software inspection
9. Recognize the non-verbal and verbal messages which signal problems within a group meeting
10. Use group process skills to effectively communicate with the simulated members of the inspections group

Manager Track Objectives

The Manager will be able to:

1. Identify the role of the manager in the Inspections Process
2. Describe the purpose of formal inspections
3. Identify the stages of the formal Inspection process
4. Cite the advantages of Inspections versus Walkthroughs
5. Describe the way in which Inspections data can be used in future software development planning
6. Identify the key features of formal inspections that contribute to cost saving and error reduction
7. Cite the studies which support the use of formal inspections within organizations
8. Describe a process for implementing formal inspections within an organization
9. Describe the key philosophical aspects of implementing inspections within an organization.

APPENDIX E

CARNEGIE MELLON CODE INSPECTION MODEL

Advanced Learning Technologies Simulation

The Inspections Process

Planning
Preparation
Entry Criteria
Conduct
Exit Criteria
Reporting
Follow-up

Defined Roles In the Simulation

Differences described in roles between the ALT project and other models are primarily due to the interface issues

Manager - receives reports, manages follow-up

Producer - produces product, satisfies entry criteria, explains product, contributes as inspector, reworks product

Moderator - verifies entry criteria via E-mail, schedules meeting via E-mail, provides meeting notice and materials (checklists) via E-mail to team, conducts overview, prepares for review as any other inspector, directs inspection, handles final disposition of the meeting, completes summary report, sends report to manager via E-mail

Recorder - prepares for review as any other inspector, records defects, records issues, raises issues, provides defect log to moderator

Reader - prepares for review as any other inspector, guides team by pacing the examination of the material, paraphrases material, raises issues

Reviewer (All 4 people listed above - producer, moderator, recorder, reader) - responsible for effective participation

Review Sequencing

The code provides the main mechanism for sequencing the review discussion. Checklists will be available for the preparation process. Entry and exit criteria will be checked during the simulation. Recording logs and summary reports will be constructed as templates.

APPENDIX F

IBM COMPARISON OF OBJECTIVES AND CODE INSPECTION MODELS

From: QRR --HQUVMS00
To: WFRUETT --VMSFFHQU

Date and time 04/10/92 12:31:50

FROM: JIM QRR,EXT8491

*** Resending note of 04/10/92 14:46

Below is the response to the two actions due by mid April. If there is any additional information that is needed, please contact me.

Should I be out of the office, please feel free to use the BEEPER 830-5950 to reach me without playing "phone tag".

Subject: Reply To Action On Instructional Tool For Inspections

Reference: 03/19/92 Meeting with FDSD, Southwest Research Institute, and IBM to demo tool developed by Carnegie Mellon using DARPA funds.

Scope: This memo completes the following:
(1) Evaluate the Carnegie Mellon Objectives
(2) Evaluate the IBM inspection process model described by Carnegie Mellon

Evaluate the Carnegie Mellon Objectives For The Code Inspection Course

1. Additional items need to be added under "Describe the purpose of formal inspections". These additional items are:
 - Inspections are performed with the expectation that all errors found will be corrected before they are propagated to further products
 - Data produced is used for process improvement (in addition to project management and process control).
2. Additional items need to be added under "Identify the stages of formal inspection process. These additional stages are:
 - Entry Criteria (prior to Planning). Inspections must satisfy a set of measurable actions that must be complete before each type of inspection can take place.
 - Post Meeting Errors (after the Reporting, in parallel with Re-work and Follow-Up. If errors are discovered in the product after the inspection meeting is complete, but prior to the product being delivered to the next step in the process, then an inspection process step is in place to capture the information on these errors the same as if they had been discovered during the inspection meeting. Errors found after the inspection meeting are either corrected prior to delivery of the product to the next step in the process, or else formal Problem Reports against the error are entered in the Configuration Management system.
 - Exit Criteria (after Re-inspection). Inspection process must satisfy a set of measurable actions before development can proceed to the next step in the software engineering process. These actions shall insure that all major errors have been corrected, or documented via formal Problems Reports.
3. Additional item needs to be added under "Describe the benefits of the formal inspection process". The item is:
 - Provide near immediate feedback to the process developing the product under inspections of escapes that are occurring. This information allows very rapid process improvement for the prior process.

4. Additional items needed to be added under "Describe the role of planning and preparation in the inspection process--Planning".
The additional items are:
 - Determine the number of parts of a product to inspect at a single inspection meeting so as to not exceed the 2 hour meeting time limit
 - Determine all inspectors whose support is essential for the success of the meeting and identify those inspectors as mandatory for the inspection meeting.
 - Inspection meetings shall be scheduled far enough in the future to allow at least the minimum lead time defined by the inspection process
5. Additional item needed to be added under "Describe the role of planning and preparation in the inspection process--Preparation".
The additional item is:
 - All errors shall be documented for presentation during the meeting.
6. Additional items needed to be added under "List the review roles assumed during the inspection". Also, one role is not required.
The role not required:
 - Recorder (Errors are documented prior to inspection meeting by inspectors, collected by moderator. Moderator responsible for producing final report, including summary of actions.).
Additional roles needed:
 - Peer
 - Representative from a previous step in the s/w engineering process (i.e., requirements analysts for design/code inspection).
 - Representative from the next step in the s/w engineering process (i.e., integration or independent test for design/code inspection).
7. Information presented under "List the types of checklists that can be used before, during the inspection and for follow-up" could be presented in a number of ways. Checklists should promote finding the following general types of errors:
 - Data
 - Requirements compliance
 - Interfaces
 - Logic
 - Standards compliance
 - Performance
 - Readability
8. Disagree with the following characteristics under "Characteristics of an inspection meeting":
 - recorder records defects on Inspection Defect Log
(would simply say that moderator is responsible for defects to be recorded on Inspection Defect Log).
9. Additional item needed to be added under "Basic rules for Code Inspections". The additional item is:
 - Team of inspectors owns the product after the inspection and are jointly responsible for all errors not detected. Quality ownership transfers from producer to inspection team at the end of the inspection meeting.
10. Disagree with item under "Module 2 Objectives, Item 2 Compare the differences between inspection and walkthrough procedures, - Software Inspections, - Disadvantages".
 - An inspection does not have to be keyed to developer's viewpoint.

Proper inspections will have other organization roles represented other than the developer.

11. Additional items needed to be added under Module 3 Objectives, Item 1.A Moderator role description. The additional items are:
 - Document and classify errors
 - Disposition errors as to action to be taken
 - Assign errors dispositioned for correction to the author
 - Verify, personally or by delegation, that all errors disposition for correction are actually corrected prior to authorizing delivery to the next step in the software engineering process.
12. Disagree with role of Recorder under Module 3, Item 1.C. These actions are performed by a combination of inspectors (records errors found in preparation) and moderator (records errors found real time in inspection meeting, collects other recorded errors, and prepares summary meeting reports).
13. In Module 3, Item 3, impacted by previous comments, especially expanded moderator role, additional roles, and deletion of recorder role.
14. In Module 3, Item 4, the data required is insufficient. Refer to Section 3.6 Required Data of the Software Inspection Process Standard produced via SQA Standards RTOP under this contract (Southwest Research Institute should have been given a copy of this document previously).
15. In Module 3, Item 6, impacted by previous comments, especially expanded moderator role, additional roles, and deletion of recorder role.
16. In Module 3, should also acknowledge the importance of inspecting software as an organizational approach to ensuring process stability/improvement.

Evaluate the Carnegie Mellon Code Inspection Model For IBM/FSD/Houston

1. Process steps adequate. Exit criteria should be last. Missing items like network, post meeting errors, collection of inspection meeting reports, submission of summary data to database, extraction of reports from database, summary metric data, FACI/CI summary data, etc.
2. Under topic "Defined Roles":
 - Manager: Multiple managers involved. Development manager assigns individual to work a particular change instrument (Change Request, CR, or Discrepancy Report (DR)). Verification and Requirements Analysis managers maintain lists that relate responsible individual to specific requirements. Librarian contacts assigned individual when inspection package is received. Manager involved only in special situations (excessive workload, illness, etc.)
 - All managers of individuals participating in the inspection receive reports on Major Errors discovered.
 - Development manager signs off as part of promotion (i.e., formal submission to configuration controlled library process via Configuration Management Data Base controlled build process).
 - Producer (author): Disagree with "does unit level testing prior to inspection". Unit testing (currently called development

**ORIGINAL PAGE IS
OF POOR QUALITY**

testing) is done after the inspection package is prepared, and in parallel with setting up the inspection, distribution of materials, and preparation for the inspection meeting by the moderator and other inspectors. Depending on the timing for a specific change/inspection, unit tests may have been executed. However, the ideal situation would be for the inspection meeting to occur prior to executing unit tests.

- Moderator: The single most important role. Primary role is to address unique issues, control the meeting, insure results are recorded, insure correction actions are assigned, insure correction actions are complete, and that the code inspected is ready for promotion via build process. Responsible to insure that the source promoted to the configuration controlled library via build process is identical to that inspected (if no correction), or the version reviewed to insure all corrections were done correctly.

NOTE: Moderator does not have to be from the organization of the Producer. Moderator Pool includes Producers, Requirements Analysts, and Independent Verifiers. Moderators must have special education.

- Librarian: Performs para technical tasks including:
 - arranges the inspection meeting time and place, including making sure inspectors and moderator can support
 - elevates any scheduling issues to moderator
 - schedules inspection room and distributes materials
 - post inspection meeting (after build inputs have been submitted), files meeting records.
 - enters key information in database
 - produces regular and special request reports from database
- Recorder: Role does not exist. Individual inspectors prepare error descriptions prior to meeting. Moderator responsible for recording issues discovered real time in meeting, plus all summary reports.
- Designer/Tester: Confusion of these roles. Generally, the Producer is the Designer, so this is not a unique role. Tester doing unit testing would normally be addressed as a Peer.
- Independent Tester: The independent detail verifier brings a "non-development/outsider" view and objectivity to the inspection process. Acts as a reviewer (May also have another role, e.g., reader).
- Consumer: For design/code inspection, this role is occasionally performed by members of the Test and Operations team. However, normally, this function is performed by the Requirements Analyst.

For design/code inspections, no manager or non-IBM contract personnel are allowed to participate in the inspection process. Note that excludes the NASA customer (FDSD). Included are IBM'ers and IBM subcontractor

personnel. Inspection process for design/code produced by subcontractor personnel or IBM personnel are subjected to identical process.

For Detail Verification Test Inspections, the consumer role is filled by the NASA customer (FDSD personnel) and other non-IBM participants like NASA (or non-IBM contractors) requirements owners (called principle function owners).

THANKS,
JIM

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX G
VALIDATION PLAN

NASA CODE INSPECTION INSTRUCTIONAL VALIDATION

VALIDATION PLAN

1. Content Validity of Materials (Content)

A. Compare Objectives (NASA and Carnegie Mellon)

Purpose: The purpose of this activity is to determine the objectives taught by the Carnegie Mellon Code Inspection Course and compare these with current NASA objectives to find similarities and differences in terms of content, specifically what objectives are missing or extraneous in the Carnegie Mellon Course.

Input: Objectives (4 sources)

1. Objective outline provided by Carnegie Mellon
2. Objective description (answers) provided by SwRI
3. Objective description provided by IBM analysts
4. Description of current code inspection process (NASA Software Inspection Process Standard, Software Formal Inspections Guidebook)

Activity: Compare NASA and Carnegie Mellon objectives

Output: Data for final report
Test instruments

B. Compare Models for Code Inspection (NASA and Carnegie Mellon)

Purpose: The purpose of this activity is to compare the model for code inspection used by the Carnegie Mellon Code Inspection Course and the model used by NASA for code inspection to find similarities and differences, specifically to determine if the two models are similar enough to make the Carnegie Mellon Course content relevant for use with NASA analysts.

Input: Carnegie Mellon and NASA objectives for code inspection as provided by Carnegie Mellon

NASA model for code inspection as provided in the following documents: NASA Software Inspection Process Standard, Software Formal Inspections Guidebook

Activity: Compare NASA and Carnegie Mellon models for code inspection

Output: Data for final report
Test instruments

2. Effectiveness of Materials (Presentation Strategy)

(3 NASA analysts use the Carnegie Mellon Code Inspection Course)

A. Knowledge of Information (Instructional Modules)

- pretest/posttest (paper/pencil test over Carnegie Mellon objectives)

Purpose: The specific purpose of this activity is to measure the effectiveness of the instructional modules and look in general at the effectiveness of this type of course for teaching the content, code inspection, as well as other similar processes. Analysts will be administered a pretest prior to interacting with the instructional modules (Carnegie Mellon Course) to determine their prior level of knowledge of code inspection. A

posttest will be administered after interacting with the instructional modules to determine how much was learned from the modules. Gains in scores from the pretest to the posttest will be examined.

- Input:** Pretest
Posttest
Demographic data
5 analysts (pretest) (not highly experienced in code inspection)
3 analysts (posttest)
- Activity:** Select 3 analysts to participate in validation (pretest)
3 analysts participate in the instructional module portion of the course
Test analysts after instructional modules (posttest)
- Output:** Selection of 3 analysts
Record of prior knowledge of code inspection
Gain in score attributed to instructional modules
- Note:** The pretest will determine from the original 5 analysts which 3 will participate in the validation process.

B. Application of Information (Simulated Code Inspection)

- posttest (paper/pencil test over Carnegie Mellon objectives)
- course feedback after practice code inspection

Purpose: The purpose of this activity is to measure the effectiveness of the simulation (simulated code inspection) as a teaching tool. After interacting with the simulation portion of the course, analysts will be given the same posttest (as used in Part A) to determine if there is any gain in their scores after participating in the simulation. In addition, comments will be noted regarding the course assessment of the analysts performance in the simulation.

- Input:** Posttest
3 analysts
- Activity:** 3 analysts participate in the simulation portion of the course
Test analysts after simulated code inspection (posttest)
- Output:** Gain score attributed to simulation
Program comments regarding performance

C. Overall Course

- analyst self-report (questionnaire) (interview)
- evaluator report (observations) (opinions)

Purpose: Upon conclusion of the Carnegie Mellon Code Inspection Course (instructional modules & simulation), analysts will be given a questionnaire asking for their subjective response to the course. Following are the purposes of the questionnaire: 1) to determine whether analysts liked this method of instruction, 2) to determine if analysts felt they learned from the course, 3) to identify areas where difficulties occurred, 4) to find out what they thought the strengths and weaknesses of the course were, 5) to determine opinions on whether weaknesses could be overcome, as well as possible suggestions to overcome them. The interview will be an extension of this line of questioning to give the analysts an opportunity to further express their opinions about the course and its potential for use by NASA analysts.

The evaluator's report will document observations of how the analyst performed during the course. Observations will include items such as problems encountered,

analyst comments, and time spent on various parts of the instruction. Subjective opinions of the observer may be included for the purpose of documenting or clarifying events occurring during the validation process.

Input: Analyst questionnaire
Analyst interview
Evaluator observations
Evaluator opinions
Activity: Administer questionnaire
Interview analyst
Observe analyst using the code inspection course
Output: Analysts' objective responses to the course
Observer's documentation of what occurred during the course

NOTE: Part 2 will take place over two days using analysts provided by NASA:

Day 1: Set up the computer system
Use materials with 1 analyst (approximately 3-4 hours)
Day 2: Use materials with 2 analysts (approximately 3-4 hours each)

NOTE: Instruments to be developed:

1. Pretest/Posttest

Purpose: The purpose of this test is to measure the effectiveness of the instructional modules and the code inspection simulation. The pretest and posttest will be the same instrument. This test will be comprised of content taken directly from the Carnegie Mellon objectives addressed in the five instructional modules of the course. Test answers will be reverse engineered from the Carnegie Mellon course by SwRI.

2. Analyst Questionnaire

Purpose: The overall purpose of the analyst questionnaire is to determine the potential for using this course and/or the feasibility of this type of instruction for code inspection or other similar processes (content) relevant to NASA needs. The questionnaire will consist of items for the analyst to respond to regarding their opinions about the Carnegie Mellon Course.

3. Interview Questions

Purpose: The purpose of the interview is to give each analyst further opportunity to express his/her opinions regarding strengths and weaknesses, and the potential for this course or this type of instruction for teaching code inspection and other content relevant to NASA.

4. Observation Form

Purpose: The purpose of the observation form is to provide additional descriptive data which may aid in explaining results found in the data collected. The observation form will be used by the observer to document any problems encountered, any comments made by the analyst either positive or negative, and how long the analyst spent during parts of the course.

Note: The observer will play an impartial role in documenting events, however, a secondary role of the observer is to facilitate analysts in staying on task to insure validity of comparisons between the three analysts involved.

5. Demographic Data Sheet

Purpose: The purpose of the demographic data sheet is to provide information which might help analyze results found in the data collected. The data sheet will be used to provide background information about the analysts participating in the validation study.

3. Final Validation Report

A. Purpose of the Study

The purpose of this study is to validate the instructional effectiveness of the Carnegie Mellon Code Inspection Course. This code inspection course validation provides a case study for exploring process simulation training with a subject matter domain of software.

B. Materials

The materials used in this study will be listed and briefly described including the Carnegie Mellon Course itself, as well as the pretest/posttest, analyst questionnaire, interview questions, observation form, and the demographic data sheet.

C. Procedures

The procedures used in collecting the data described in Part E (Results) will be stated.

D. Subjects

The subjects (analysts) participating in this study will be briefly described. This description will give general information about the subjects, based on the information they provide on their demographic data sheets, as well as any observations made by observers during the validation process.

E. Results

Results will be reported for two areas including content and instructional effectiveness. Results will be presented based on the data gathered by the instruments described previously.

1) Content Data

- a) Objectives
- b) Models for Code Inspection

2) Instructional Effectiveness Data

- a) Pretest/Posttest (Knowledge of Information)
- b) Posttest (Application of Information)
- c) Course feedback after practice code inspection (Application of Information)
- d) Analyst Self-Report (questionnaire, interview)
- e) Evaluator Report (observations, opinions)

F. Conclusions and Recommendations

SwRI will synthesize and interpret the results and summarize the subjective conclusions. SwRI will provide a professional recommendation based on these results and conclusions. Recommen-

dations will be made as to the use of this specific course, as well as this type of instruction in general.

G. Limitations

Any factors seen as limitations on this study which may affect the results will be clearly stated so that knowledge of this information can be used in interpreting the results. One example of such a limitation is that the scope of this project does not allow for a true empirical study to be implemented. This is not possible with the limitation of only three analysts using the course. Due to the small number of subjects, statistical manipulation of the data is not appropriate, therefore data gathered will be descriptive in nature.

H. Appendix

The appendix will contain copies of all pertinent documents to this study. Such documents will include instruments, Carnegie Mellon and NASA objectives, Carnegie Mellon and NASA models for code inspection, and any other documents relevant to the validation process.

APPENDIX H
OUTLINE OF "A CURE FOR THE COMMON CODE"

NASA CODE INSPECTION INSTRUCTIONAL VALIDATION

STRUCTURE OF THE CODE INSPECTION COURSE SOFTWARE:

Auditorium

- motivational presentation (why software quality is important and how inspections improve the quality of software)
- information on the company Ultimex and course the user is about to take

Training Room

- learn how to navigate through the simulated world
- access to instruction on inspections and group processes

Instructional Modules (5)

1. Formal Inspections: Purpose and Process

The Purpose (4 minutes)

The Process (9 minutes)

The Process (3 minutes)

Conducting An Inspection (3 minutes)

Guidelines For Success (3 minutes)

2. Inspection Types and Differences

Inspections

Definition

Function

Walkthroughs

Definition

Function

Reviews

Definition

Function

Advantages/Disadvantages

Inspections

Walkthroughs

Reviews/Audits

Quality Assurance Techniques

3. Inspection Roles and Pitfalls

Moderator

Role (14 minutes)

Problems (up to 10 minutes)

Attack On Producer (2 minutes)

Moderator Dominates (2 minutes)

Producer Problem Solving (2 minutes)

Follow-Up Communications (2 minutes)

Pitfalls (2 minutes)

Helpful Hints (up to 20 minutes)

What if: (20 items)

Checklists (up to 10 minutes)

Before

During

After

Producer

Role (5 minutes)

- Problems (up to 6 minutes)
 - Attack On Producer (2 minutes)
 - Producer Problem Solving (2 minutes)
 - Pitfalls (2 minutes)

Reader

- Role (3 minutes)
- Problems (up to 4 minutes)
 - Reader Too Fast (2 minutes)
 - Pitfalls (2 minutes)

Recorder

- Role (7 minutes)
- Problems (up to 5 minutes)
 - Recorder Too Slow (3 minutes)
 - Pitfalls (2 minutes)

4. Inspection Tools and Forms (practice code inspection)

- How To Get Help (2 minutes)
- Office Window Environment (3 minutes)
- Tools For Code Analysis (up to 9 minutes)
 - Hypertext Tools Demo (4 minutes)
 - Code Debugger Tools Demo (5 minutes)
- Electronic Mail (not available)
- Practice With The Tools

5. Inspection Communications

- Group Process (up to 23 minutes)
 - What Is A Group? (5 minutes)
 - Different Types of Groups (5 minutes)
 - Group Communication (5 minutes)
 - Special Problems (8 minutes)
- Conversational Interface (6 minutes)

Library

- contains text, graphics, video, and audio materials
- Videotapes (11 available)
- Card Catalog
 - Forms
 - Style Manuals
 - Articles
 - Slides (not available)

User's Office

- examine code that will be the subject of a later inspection
- purpose is to teach the importance of preparation in an inspection and to give the student experience in preparing for an inspection
- tools available to help prepare for the inspection (hypertext system, source level debugger)
- secretary asks user to choose role for the inspection (moderator, reader, recorder)

Conference Room

- location for the simulated code inspection
- code, specifications and error report forms complete with hypertext links are available here during the simulation of the inspection

Secretary

- user does not have access to this room

Coffee Room

- upon conclusion of the inspection, the user is given performance feedback here

Copies of this publication have been deposited with the Texas State Library in compliance with the State Depository Law.
